

LAPORAN LATIHAN ILMIAH TAHAP AKHIR

WXES 3182

Perpustakaan SKTM

SISTEM PENGESANAN DAN PEMBETULAN RALAT MENGGUNAKAN TEKNIK PENGKODAN HAMMING DAN PEMERIKSAAN LEWAHAN BERKITAR(CRC) / MODUL PENGEKODAN

OLEH:

FAUZIAH MAHMOD

WEK000423

Laporan ini merupakan sebahagian daripada keperluan Ijazah Sarjana Muda

Sains Komputer dan Teknologi Maklumat Universiti Malaya

SESI 2002/2003

Penghantaran maklumat dari satu destinasi ke destinasi yang lain melibatkan konsep kebolehppercayaan dan intergriti maklumat tersebut. Kadang-kala maklumat yang dihantar adalah tidak sama dengan maklumat yang diterima. Penghantaran maklumat dalam bentuk bit antara satu komputer dengan komputer yang lain juga mengalami masalah yang sama. Oleh kerana data dihantar dalam susunan bit, susunan bit asal adalah tidak sama dengan susunan bit yang diterima. Maka data tersebut dikira rosak. Pelbagai kaedah untuk menjamin penghantaran data tersebut selamat. Antaranya ialah penggunaan kaedah pengekodan Hamming dan pemeriksaan lewahan berkisar (CRC).

Litar pengekodan dan penyahkodan kod Hamming dan CRC akan menjalankan fungsi tersebut. Litar ini akan mengesan sekiranya ralat terjadi semasa penghantaran semasa dihantar oleh komputer penghantar kepada komputer penerima. Ia dapat dijalankan dengan kewujudan beberapa bit lewahan yang berfungsi sebagai penunjuk jika ralat berlaku. Kajian keseluruhan akan dijalankan ke atas kedua-dua cara pengekodan ini merangkumi pengesanan dan pembetulan ralatnya.

2.2.3	Proses pengekodan	13
2.3	Kaedah Pemeriksaan Lewahan Berkisar	14
2.3.1	Polinomial	15
2.3.2	Operasi pengekodan CRC	15
2.4	Perbincangan Kod-Kod Linear	17
3.0	Metodologi	21
3.1	VHDL	21
3.2	Web Pack 4.2	25
3.3	Keperluan Sistem	30
4.0	Aliran Raksabentuk	31
4.1	Diagram Arus Tertinggi	31
4.2	Diagram Arus Tertinggi Model Pengekod	32
4.3	Pengekodan Hamming	33
4.3.1	Operasian Kod Hamming	34

Bil	Tajuk	mukasurat
4.5	Litar Gabungan	38
5.0	Abstrak	ii
5.1	Kandungan	iii
5.1.1	Penghargaan	v
5.1.2	Senarai Jadual	vi
5.2	Senarai Rajah	vii
1.0	Pengenalan	1
1.1	Skop	1
1.2	Objektif	3
1.3	Kekangan	3
1.4	Perancangan Aktiviti	5
2.0	Kajian Literasi	9
2.1	Tujuan Pengekoden	9
2.2	Kod Hamming	9
2.2.1	Bit Pariti Hamming	10
2.2.2	Matriks penjana	12
2.2.3	Proses pengekoden	13
2.3	Kaedah Pemeriksaan Lewahan Berkitar	14
2.3.1	Polinomial	15
2.3.2	Operasi pengekoden CRC	15
2.4	Perbincangan Kod- Kod Linear	17
3.0	Metodologi	21
3.1	VHDL	21
3.2	Web Pack 4.2	25
3.3	Keperluann Sistem	30
4.0	Aliran Rekabentuk	31
4.1	Diagram Aras Tertinggi	31
4.2	Diagram Aras Tertinggi Modul Pengekod	32
4.3	Pengekoden Hamming	33
4.3.1	Operasian Kod Hamming	34

4.4	Pengekodaan CRC	35
4.5	Litar Gabungan	38
5.0	Perlaksanaan	39
5.1	Perlaksanaan Fasa 1	39
5.1.1	Teknik Pengekodaan Hamming	40
5.1.2	Teknik Pemeriksaan Lewahan Berkitar (CRC)	41
5.2	Fasa 2: Penggabungan Modul CRC dan Hamming	42
5.2.1	Modul Multipleksor Hamming	43
5.2.2	Modul Multipleksor CRC	44
5.2.3	Modul 1: Penggabungan Pengekod dan Penyahkod Hamming	45
5.2.4	Modul 2: Penggabungan Modul Pengekod dan Penyahkod CRC	48
6.0	Pengujian Sistem	52
6.1	Pengujian Submodul Hamming	52
6.2	Pengujian Submodul CRC	56
6.3	Pengujian Submodul Multipleksor Hamming	60
6.4	Pengujian Modul Multipleksor CRC	63
6.5	Pengujian Modul Hamming	65
6.6	Pengujian Modul CRC	68
7.0	Perbincangan	73
7.1	Perubahan Rekabentuk Awal Sistem	73
7.2	Kelemahan dan Kekurangan Sistem	75
7.3	Perbandingan Modul Hamming dengan Sistem Sebelumnya	76

PENGHARGAAN

Setinggi ucapan terima kasih diucapkan kepada En Mohd Yamani Idna bin Idris selaku penasihat tesis saya yang banyak memberi panduan semasa saya menjalankan latihan ilmiah ini. Tidak dilupakan juga kepada modurator iaitu En. Zaidi Razak yang banyak memberi idea yang bernas dalam memperbaiki kualiti kajian ilmiah ini. Rakan sekumpulan, Wan Ramlah binti Wan Ibrahim yang banyak membantu dalam proses pencarian maklumat. Ibubapa, En Mahmud bin Md. Farip dan Puan Jamiah Baba yang berperanan besar memberikan sokongan moral kepada saya untuk terus berusaha dan tidak ketinggalan rakan – rakan seperjuangan yang lain.

Sokongan yang diberikan oleh semua pihak yang disebutkan di atas membolehkan saya meneruskan kajian ilmiah ini dan akan diteruskan dengan latihan ilmiah dua pada semester hadapan.

Pengalaman mempelajari sendiri apa yang tidak diketahui menjadikan saya lebih berdikari dan lebih bersedia menghadapi masa akan datang terutama dalam menghadapi alam pekerjaan sebenar.

Diharapkan agar kusus latihan ilmiah ini diteruskan supaya lebih banyak penemuan baru ditemui yang mungkin berguna apabila berada di alam pekerjaan kelak.

SENARAI JADUAL

BIL	TAJUK	m/s
1.0	Garis Masa untuk WXES 3181	5
1.1	Jadual Perancangan WXES 3181	6
1.2	Jadual Perancangan WXES 3182	8
1.3	Garis Masa WXES 3182	9
2.0	Bit Maklumat Hamming	11
2.1	Kedudukan Bit Pariti dan Persamaan Polinomial	15
6.0	Jadual Kebenaran Pengekodan Hamming	55
6.1	Jadual Kebenaran CRC	59
7.0	Perbandingan Antara Sistem Pengesanan dan Ralat (B) dan Sistem Pengesanan dan Penyahkod Ralat (A).	77

4.4	Litar Pengekod CRC	35
4.5	Litar Gabungan Pengekod	38
6.0	Simulasi Submodul Pengekod Hamming	54
6.1	Simulasi Submodul Pengekod CRC	58
6.2	Simulasi Submodul Multiplexor Hamming	62
6.3	Simulasi Modul Hamming	63
6.4	Simulasi Modul CRC	71
7.0	Lebariti simulasi diagram aras tertinggi untuk keseluruhan sistem	74

SENARAI RAJAH

BIL	TAJUK	m/s
2.0	Ilustrasi Titik Dimensi Hamming	10
2.1	Kedudukan Mesej Bit Terkod	12
2.2	Proses Pengekodan CRC	14
2.3	Lakaran Kasar Litar Pengekod CRC	16
3.0	Pengaliran Proses dalam VHDL	21
4.0	Rajah Diagram Aras Tertinggi	31
4.1	Blok Diagram Pengekodan Data	32
4.2	Litar Hamming	33
4.3	Litar Pengekod Hamming	34
4.4	Litar Pengekod CRC	35
4.5	Litar Gabungan Pengekod	38
6.0	Simulasi Submodul Pengekod Hamming	54
6.1	Simulasi Submodul Pengekod CRC	58
6.2	Simulasi Submodul Multipleksor Hamming	62
6.3	Simulasi Modul Hamming	63
6.4	Simulasi Modul CRC	71
7.0	Lakaran semula diagram aras tertinggi untuk keseluruhan sistem	74

1.0. PENGENALAN

Proyek ini terbagi kepada dua modul yaitu pengkodean dan penyinkronan. Untuk modul pengkodean, definisi masalah yang telah didapatkan adalah bagaimana untuk menghasilkan 7 bit enkripsi yang terdapat dari 4 bit informasi yang dimasukkan menggunakan teknik pengkodean Hamming dan CRC serta mencari komponen-komponen yang sesuai untuk dimasukkan dalam liter tersebut. Selain daripada itu perlu mencari tujuan dan objektif projek ini. Oleh kerana projek ini hanya dijalankan sebagai projek akhir semester maka pastikan yang sesuai perlu diketahui dan kriteria yang diperlukan untuk projek tersebut membolehkan definisi projek ini. Untuk modul penyinkronan, akan mencari cara yang dilakukan untuk menyinkronkan data yang diterima yang diintegrasikan dalam projek ini. Modul penyinkron akan mencari kedudukan relatif dan juga boleh menentukan relatif tersebut.

1.1. SNOP

Rajuan di atas ini menunjukkan pengiraan dan pembetulan ralat yang dilakukan di dalam sistem komunikasi digital menggunakan teknik CRC. Bahagian yang diintegrasikan adalah pengkodean 4 bit informasi kepada 7 bit enkripsi. Oleh kerana cara untuk pengkodean digunakan ialah kod Hamming dan pemeriksaan kesalahan berbilang maka kedua-dua jenis kod ini akan dikaji.

Bab pertama akan menunjukkan pengenalan kepada projek, masalah-masalah yang perlu diselesaikan, skop permasalahan yang perlu diselesaikan dan kekangan-kekangan terhadap pembinaan sistem. Kekangan yang dimaksudkan bukanlah dari segi

1.0 PENGENALAN

Projek ini terbahagi kepada dua modul iaitu pengekodan dan penyahkodan. Untuk modul pengekodan, definisi masalah yang telah dikenalpasti adalah bagaimana untuk menghasilkan 7 bit maklumat yang terkod dari 4 bit maklumat yang dimasukkan menggunakan teknik pengekodan Hamming dan CRC serta mencari komponen-komponen yang sesuai untuk dimuatkan dalam litar tersebut. Selain daripada itu perlu mencari tujuan komponen tersebut dan aplikasinya. Oleh kerana projek ini hanya dijalankan sehingga peringkat simulasi sahaja, maka perisian yang sesuai perlu ditentukan dan kriteria yang diperlukan untuk perisian tersebut mestilah memenuhi objektif projek ini. Untuk fungsi penyahkodan pula, ia akan menerima mesej yang dihantar. Ia dibangunkan secara berasingan daripada modul yang dibangunkan dalam projek ini. Modul penyahkod akan mengesan kedudukan ralat dan juga boleh membetulkan ralat tersebut.

1.1 SKOP

Kajian ilmiah ini merangkumi pengesanan dan pembetulan ralat yang dilakukan di dalam satu litar yang menggabungkan kaedah Hamming dan CRC. Bahagian yang dibangunkan adalah pengekodan 4 bit maklumat kepada 7 bit maklumat. Oleh kerana dua cara untuk pengekodan digunakan iaitu kod Hamming dan pemeriksaan lewahan berkisar maka kedua-dua jenis kaedah pengekodan ini akan dikaji.

Bab pertama akan merangkumi pengenalan kepada tajuk, masalah - masalah yang perlu diselesaikan, skop permasalahan yang perlu diselesaikan dan kekangan - kekangan terhadap pembinaan sistem. Kekangan yang dimaksudkan bukanlah dari segi

peralatan atau kewangan sebaliknya adalah kekangan yang dihadapi ketika mencari nilai teori yang betul untuk dikaitkan dengan komponen litar yang digunakan.

Bab dua merangkumi kajian literasi yang melakukan perincian tentang isu yang dikupas dalam bab pertama. Kebaikan dan kelemahan kedua-dua kaedah pengkodan ini dan aplikasinya juga akan dikaji dan akan dibandingkan dengan kaedah pengkodan yang lain.

Bab ketiga akan menjelaskan tentang metodologi yang digunakan untuk tajuk ini. Oleh kerana modul pengkodan ini dijalankan dengan kebaikan menggunakan VHDL sebagai bahasa pengaturcaraan akan dibincangkan.

Bab yang keempat adalah peringkat rekabentuk di mana semua modul yang terdapat dalam sistem akan dilakarkan secara terperinci dalam bentuk blok diagram yang kemudiannya akan diimplemetasikan di dalam perisian WebPack 4.2 . Carta alir, gambarajah blok asas, gambarajah blok yang terperinci dan rajah litar akan disertakan sebagai alat bantu dalam menjelaskan lagi rekabentuknya .

Bab yang kelima pula menyentuh tentang pelaksanaan sistem. Semua peringkat rekabentuk modul- modul dan submodul- submodul yang telah dibangunkan pada fasa sebelumnya akan dibangunkan dan ditukarkan ke dalam bentuk pengaturcaraan VHDL.

Bab yang ke enam menerangkan bagaimana pengujian dilakukan ke atas kod pengaturcaraan . Bit – bit ujian akan dimasukkan ke dalam setiap modul dan submodul untuk memastikan setiap komponen tersebut berfungsi seperti yang dikehendaki.

Bab tujuh adalah bab yang terakhir yang membincangkan tentang keputusan yang diperolehi daripada pengujian yang dilakukan, kelebihan dan kekurangan sistem, peningkatan yang boleh dijalankan pada masa hadapan dan kesimpulan bagi projek yang telah dibangunkan. Perubahan – perubahan dari segi rekabentuk sistem dan

penggunaan perisian yang digunakan untuk membangunkan projek ini juga akan dibincangkan.

1.2 OBJEKTIF

Objektif kajian yang dijalankan hanya merangkumi bahagian pengkodan kod Hamming dan pemeriksaan lewahan berkisar sahaja.

- Menukarkan nilai empat bit maklumat kepada tujuh bit maklumat.
- Menselarikan nilai teori yang didapati dengan nilai sebenar yang didapati.
- Mengetahui kebaikan penggunaan VHDL berbanding cara pengkodan yang lain.
- Memahami kod Hamming dan CRC tersebut dengan terperinci.
- Boleh membandingkan kelemahan dan kebaikan kod Hamming berbanding dengan kaedah pengkodan yang lain.
- Dapat merekabentuk litar yang menggabungkan kedua-dua kaedah pengkodan tersebut.

1.3 KEKANGAN YANG DIHADAPI

Kekangan yang dihadapi akan dibincangkan dari segi:-

- Rekabentuk litar yang dihasilkan pada peringkat ini adalah berdasarkan nilai teori semata-mata, maka perlaksanaannya pada fasa dua akan mungkin akan menghasilkan ralat.
- Penerangan secara teoritikal yang amat menjurus kepada kaedah matematik, apabila melibatkan dimensi (contohnya :matriks), menyebabkan ia agak sukar

1.4 difahami. Seseengah implementasi komponen mungkin hanya difahami perlaksanaannya melalui penerangan yang tidak melibatkan rumus matematik .

- Kekurangan masa mungkin akan menyebabkan rekabentuk litar secara terperinci tidak dapat dihasilkan. Tetapi ia akan diperbaiki pada peringkat implementasinya.
- Penggunaan perisian WEB PACK 4.2, mungkin akan ditukar perisian lain yang dirasakan sesuai, ini kerana terdapat beberapa kelemahan dalam perisian tersebut. Antaranya adalah tarikh tamat tempoh perisian.

	Diagnosis tugas antara ahli kumpulan
3	Mencari bahan rujukan melalui internet, bahan bacaan
4	Mencari bahan rujukan, melalui internet, bahan bacaan
5	Mula menulis tesis untuk bab pertama
6	Mencari bahan bab kedua
7	Mula mencari bahan untuk bab kedua
8	Memasang pertalan yang sepatutnya yang dapat melarikan program atavara. Mula menulis bab kedua
9	Menulis bab kedua : mengkaji dengan lebih mendalam tentang kaedah yang digunakan
10	Menulis bab kedua : mencari perbandingan antara teknik yang digunakan dengan teknik yang lain, VIVA
11	Mencari bahan untuk bab ketiga

1.4 PERANCANGAN AKTIVITI

Jadual 1.0 : Garis masa untuk WXES 3181

Minggu	Aktiviti
1	Memilih tajuk tesis
2	Memahami tajuk tesis kenalpasti objektif ,skop dan pengenalan projek Bahagikan tugas antara ahli kumpulan
3	Mencari bahan rujukan melalui internet, bahan bacaan
4	Mencari bahan rujukan, melalui internet, bahan bacaan
5	Mula menulis tesis untuk bab pertama
6	Mencari bahan bab kedua
7	Mula mencari bahan untuk bab kedua
8	Memasang perisian yang sepatutnya yang dapat melarikan program aturcara. Mula menulis bab kedua
9	Menulis bab kedua : mengkaji dengan lebih mendalam tentang kaedah yang digunakan
10	Menulis bab kedua : mencari perbandingan antara teknik yang digunakan dengan teknik yang lain, VIVA
11	Mencari bahan untuk bab ketiga

Jadual 1.1 :Jadual perancangan WXES 3181

Tugas/Minggu	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Pilih tajuk	■														
Bahagi tugas		■													
Cari bahan			■	■	■										
Bab pertama					■										
Cari bahan						■	■								
Perisian								■	■						
Bab kedua								■	■	■					
Cari bahan											■				
Bab ketiga												■			
Cari bahan													■		
Bab keempat														■	
Penyemakan															■
Penghantaran															■

Jadual 1.0 dan 1.1 menunjukkan proses pengurusan masa untuk projek pengkodan dan pembedulan ralat . Masa yang diberikan untuk menyelesaikan kajian ilmiah fasa pertama ini adalah selama lima belas minggu.. Ia dijalankan pada semester pertama. Sebarang perubahan terhadap kajian yang dilakukan akan dapat dilakukan selepas minggu ke sepuluh iaitu selepas sesi “viva” dilaksanakan.

Jadual 1.2 : Jadual perancangan WXES 3182

1	Pemeriksaan semula perisian yang sesuai untuk kaedah yang digunakan
2	Proses rekabentuk pengekodan
3	Proses rekabentuk pengekodan
4	Merangka bab pertama
5	Proses pelaksanaan
6	Merangka dan penulisan bab kedua
7	Proses pengujian
8	Merangka bab ketiga
9	Mencari bahan untuk perbincangan projek
10	Mencari bahan untuk perbincangan projek
11	Penyediaan dokumentasi lengkap projek
12	Penyediaan dokumentasi lengkap projek
13	Penyediaan dokumentasi lengkap projek
14	Penghantaran WXES 3182

Jadual 1.3: Garis masa WXES 3182

Tugas/Minggu	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Perisian														
Pengekodan														
Bab pertama														
Perlaksanaan														
Bab dua														
Pengujian														
Bab ketiga														
Perbincangan														
Dokumentasi														
Penghantaran														

Jadual 1.2 dan 1.3 , menunjukkan perancangan untuk latihan ilmiah dua atau WXES 3182. Ia dijalankan pada semester kedua. Untuk kajian ilmiah ini, ia adalah lebih menjurus kepada pengujian ke atas rekabentuk yang dijalankan melalui proses simulasi.

2.0 KAJIAN LITERATUR

Bab ini akan membahas tentang pengenalan kepada kod Hamming, kod pemeriksaan kesalahan berkitar (CRC) dan perbandingan antara kedua-dua jenis kod ini dengan "non-cyclic codes". Kod Hamming dan CRC berfungsi bermaksud operasi penambahan modulo dua yang operasinya sama dengan get XOR.

2.1 TUJUAN PENULISAN

Tujuan penulisan ini adalah untuk memberi pengetahuan kepada komputer penerima bagaimana cara mengkodkan data yang akan dikirimkan kepada komputer penerima. Apabila komputer penerima menerima data yang diterima, maka komputer penerima akan memeriksa data tersebut apakah data tersebut benar atau salah. Terdapat beberapa jenis ralat yang ialah ditakrifkan ialah:

- Ralat tunggal
- Ralat dua bit
- Ralat berbilang

KAJIAN LITERASI

2.2 KOD HAMMING

Kod Hamming ialah kaedah pengesanan dan pembetulan ralat berdasarkan bit parity. Kaedah ini boleh mengesan sehingga dua bit ralat dan membetulkan satu bit ralat.

2.0 KAJIAN LITERASI

Bab ini akan membincangkan tentang pengenalan kepada kod Hamming, kod pemeriksaan lewahan berkisar (CRC) dan perbandingan antara kedua-dua jenis kod ini dengan "convolutional codes". Kod Hamming dan CRC berfungsi berasaskan operasi penambahan modulo dua yang operasinya sama dengan get XOR.

2.1 TUJUAN PENGKODAN

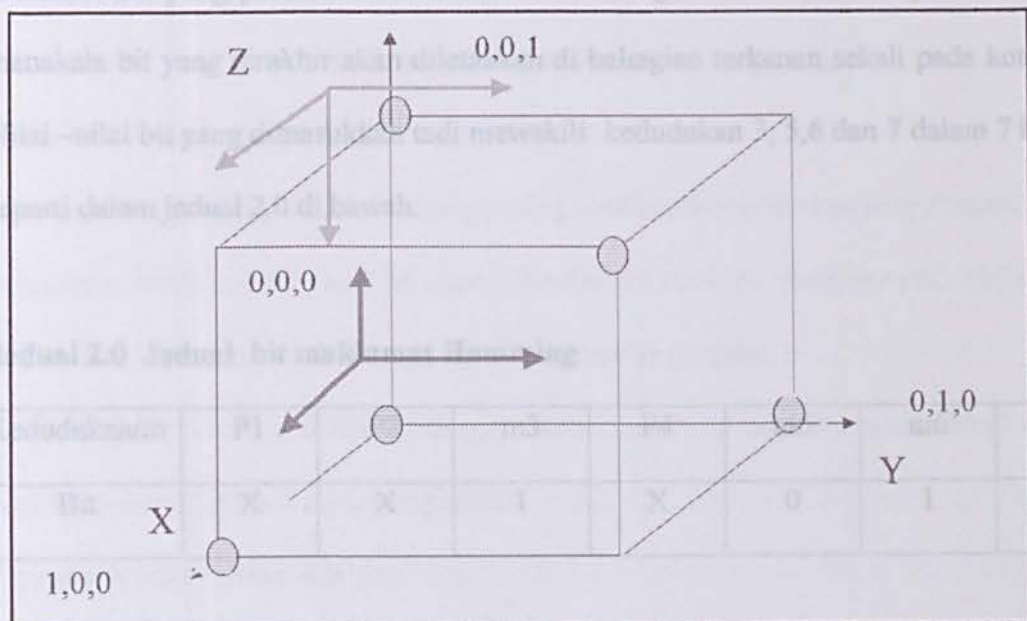
Tujuan pengkodan adalah untuk memberi garis piawai kepada komputer penerima bagaimana corak mesej yang sebenarnya. Kedah pengkodan yang digunakan oleh komputer penghantar telah dipersetujui oleh komputer penerima. Apabila komputer penerima menerima mesej tersebut ia akan berupaya mengesan sekiranya berlakunya ralat. Terdapat beberapa jenis ralat yang telah ditakrifkan iaitu:-

- Ralat bit tunggal
- Ralat dua bit
- Ralat berbilang

Ralat ini mungkin terjadi disebabkan oleh hingar semasa penghantaran.

2.2 KOD HAMMING

Kod Hamming adalah kaedah pengesanan dan pembetulan ralat berasaskan bit pariti. Kaedah ini boleh mengesan sehingga dua bit ralat dan membetulkan satu bit ralat serentak.



Rajah 2.0 Rajah ilustrasi titik dimensi Hamming

Rajah 2.0 menunjukkan bagaimana perwakilan bit pariti berdasarkan titik dimensi yang diilustrasikan oleh R. W Hamming , iaitu pengasas untuk pengkodan Hamming . Kedudukan koordinat 001, 010, 100, dan 000 diunjurkan kepada paksi X, Y, dan Z mewakili bit- bit pariti 1, 2, dan 4. Unjuran paksi –paksi ini juga merangkumi semua kedudukan koordinat yang lain dalam 7 bit data yang telah dienkodkan.

2.2.1 BIT PARITI HAMMING

Untuk kaedah pengkodan (7, 4) ini, kaedah pengesanan dan pembetulan ralat menggunakan bit- bit pariti 1,2, dan 4. Sebagai contoh, nilai bit maklumat yang ingin dihantar ialah 1011, maka untuk mengira nilai pengkodannya ialah seperti yang diterangkan di sebelah.

Nilai yang dihantar adalah 1011, maka gambarkan bahawa nilai bit yang pertama adalah 1. Bit yang pertama akan diletakkan dibahagian terkanan iaitu pada kotak m7, manakala bit yang terakhir akan diletakkan di bahagian terkanan sekali pada kotak m3. Nilai –nilai bit yang dimasukkan tadi mewakili kedudukan 3, 5,6 dan 7 dalam 7 bit data seperti dalam jadual 2.0 di bawah.

Jadual 2.0 Jadual bit maklumat Hamming

Kedudukan/m	P1	P2	m3	P4	m5	m6	m7
Bit	X	X	1	X	0	1	1

Nilai bit pariti yang dikira adalah hasil penambahan modulo dua atau operasi XOR ke atas bit- bit pada kedudukan 3,5,6 dan 7. Penambahan bit pariti ke dalam jujukan mesej m akan menghasilkan “codeword” atau C. Maka nilai bit- bit pariti akan dikira pada kedudukan X ialah:-

$$\text{Bit pariti 1, } p1 = m3 + m6 + m7$$

$$= 1 + 1 + 1 \text{ (penambahan XOR, nilai ganjil)}$$

$$= 1$$

$$\text{Bit pariti 2, } p2 = m3 + m5 + m6$$

$$= 1 + 0 + 1 \text{ (penambahan XOR, nilai genap)}$$

$$= 0$$

$$\text{Bit pariti 4, } p4 = m5 + m6 + m7$$

$$= 0 + 1 + 1 \text{ (penambahan XOR, nilai genap)}$$

$$= 0$$

Cara penyusunan “codeword” yang terhasil adalah seperti yang ditunjukkan di sebelah.

P1	P2	P4	m3	m5	m6	m7
----	----	----	----	----	----	----

Rajah 2.1 : Rajah kedudukan bit mesej terkod

Maka nilai mesej terkod , C ialah = 1001011.

Kaedah ini hanyalah kaedah yang paling asas untuk melakukan pengekodan. Dalam kehidupan sebenar, kod –kod ini akan dihantar secara blok yang banyak. Maka untuk aplikasi ini Hamming telah memperkenalkan matriks penjana.

Nilai pengekodan dengan menggunakan kaedah ini adalah mendarabkan matriks penjana tersebut dengan mesej empat bit.

Persamaan yang terlibat adalah:-

$$C = mG$$

C = 7 bit mesej yang telah dikodkan

m = 4 bit input yang hendak dikodkan

G = matriks penjana

2.2.2 MATRIKS PENJANA

Oleh kerana kod Hamming adalah kod linear, maka apabila proses penambahan modulo dua dilakukan ke atas beberapa “codeword” akan menghasilkan “codeword”.

Nilai bit maklumat adalah semua nilai yang mungkin dari 0000 sehingga 1111. Matriks penjana adalah matriks (n-k, n) , di mana n adalah nilai 7 bit terkod dan k adalah 4 bit input yang hendak dikodkan. maka untuk tujuan pengekodan (7,4), nilai matriks 3 X 7 diperolehi ,

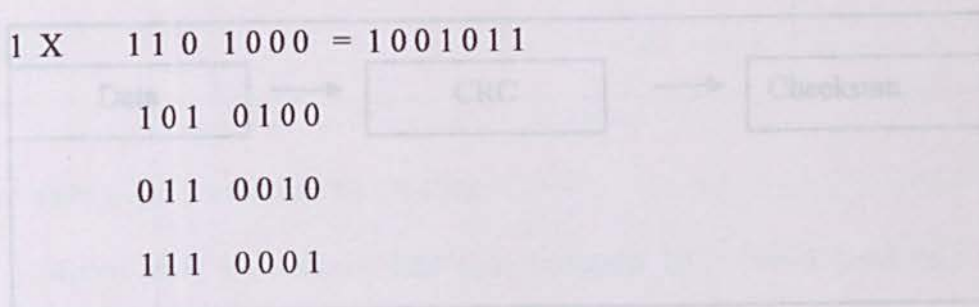
$$G = [P | I]$$

Nilai I (3 X 4) adalah nilai bit – bit mesej yang hendak dikodkan yang mewakili kedudukan 3,6, 7 dan 5 . Bit- bit ini ditulis menggunakan perwakilan binari. Nilai I pula adalah matriks identity (3 X3). Langkah pembentukan matriks penjana ini tidak akan dijelaskan dengan lebih lanjut, kerana aplikasi ini tidak akan digunakan di dalam pembangunan sistem . sebaliknya sistem akan dibangunkan berasaskan operasi modulo 2 yang akan diimplementasi menggunakan get XOR.

2.2.3 PROSES PENGEKODAN

Seperti yang telah dinyatakan di atas $C = mG$, atau “codeword” adalah hasil daripada pendaraban mesej dengan matriks penjana. Bit maklumat yang telah digunakan untuk proses pengekodan nilai 1011 akan digunakan semula. Contoh matriks penjana adalah seperti yang tertera di bawah.

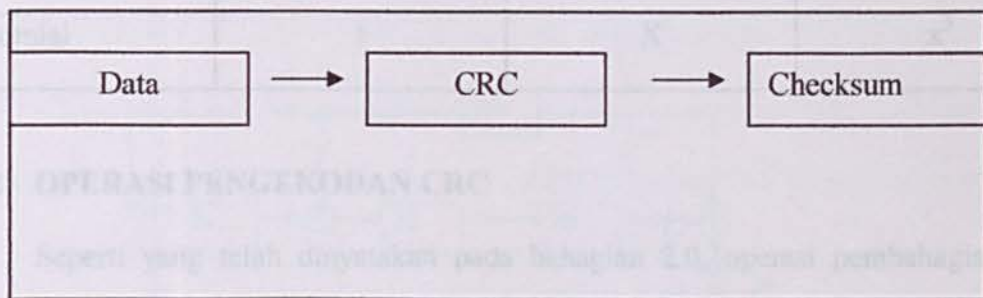
$$1011 \times 1101000 = 1001011$$



Selain daripada itu, proses pengekodan juga boleh menggunakan implementasi litar. Seperti yang diketahui, hasil penambahan modulo dua adalah sama dengan operasi implelementasi menggunakan XOR.

2.3 KAEDAH PEMERIKSAAN LEWAHAN BERKITAR

Pemeriksaan lewahan berkitar atau “*cyclic redundancy bit*” adalah satu cara untuk mengesan ralat menggunakan kaedah “*checksum*”. Ia juga berasaskan bit pariti tapi mewakili bit tersebut dengan menggunakan polinomial. Terdapat kelebihan kod ini berbanding kod Hamming kerana ia dapat mengesan nilai ralat yang lebih banyak daripada kod Hamming. Kaedah ini juga sama seperti kaedah Hamming yang memindahkan data secara blok, tetapi blok data tersebut tidak dipanggil blok, sebaliknya dipanggil kerangka. Beberapa jujukan N akan ditambah ke rangka semasa penghantaran maklumat berlaku. N bit ini adalah nilai bit pariti yang diperolehi hasil pembahagian antara bit maklumat dengan polinomial. Terdapat beberapa jenis kod CRC iaitu yang menghasilkan pengekodan dari input 4,8,16 dan 32 bit. CRC ini biasanya digunakan untuk pemindahan data internet.



Rajah 2.2 : Proses pengekodan CRC

Rajah 2.2 menunjukkan proses yang berlaku dalam proses penghasilan “checksum”. Pada mulanya 4 bit input data akan dimasukkan. Katakan nilai input yang dimasukkan adalah 1011. Dalam bentuk persamaan, ia mewakili nilai $1 + x^2 + x^3$. Persamaan tersebut akan didarabkan dengan x^3 , iaitu nilai darjah polinomial tertinggi yang terdapat dalam persamaan tersebut.

1. Bit maklumat tersebut akan dibahagikan dengan persamaan polinomial yang mewakili bit-bit pariti 1, 2, dan 4 iaitu $1 + x + x^3$. Nilai baki yang terdapat semasa operasi pembahagian akan ditambah ke bit maklumat asal dan akan dihantar ke komputer penerima.

2.3.1 POLINOMIAL

Persamaan polinomial $1 + x + x^3$ mewakili nilai kedudukan bit pariti. Secara kasarnya pengkodan menggunakan kaedah polinomial ini boleh diperolehi dengan mudah. Nilai kedudukan bit adalah sama seperti darjah untuk komponen dalam polinomial tersebut seperti yang ditunjukkan oleh jadual 2.1 di bawah.

Jadual 2.1 : Kedudukan bit pariti dan persamaan polinomial

Bit pariti	1	2	4
Polnomial	1	X	x^3

2.3.2 OPERASI PENGEKODAN CRC

Seperti yang telah dinyatakan pada bahagian 2.0, operasi pembahagian juga mempunyai nilai yang sama dengan operasi modulo dua. Oleh kerana itu, implementasi get XOR juga digunakan. Nilai pengkodan boleh diperolehi dengan mendarab bit maklumat dengan polinomial penjana. Polinomial penjana diwakili dengan darjah tertinggi untuk persamaan yang hendak dikodkan.

Misalnya nilai data yang dimasukkan ialah 1011, maka polinomial yang diwakili adalah $1+x^2+x^3$ langkah untuk mendapatkan nilai pengkodan adalah seperti di sebelah.

1. Darabkan persamaan yang hendak dikodkan dengan darjah tertinggi untuk persamaan tersebut. Oleh kerana nilai darjah tertinggi ialah 3, tambahkan 3 bit 0 pada perwakilan binarinya.

$$1+x^2+x^3 \cdot x^3 = x^6 + x^5 + x^3$$

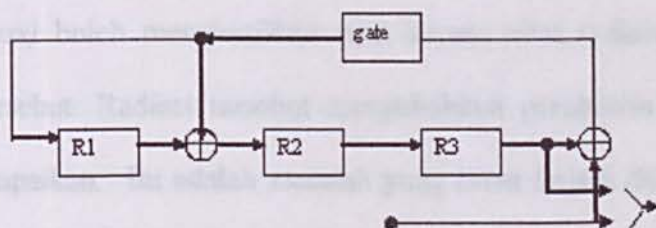
$$= u = 1011000_2$$

2. Bahagikan nilai persamaan yang didapati dengan persamaan polinomial pariti $1+x^2+x^3$ yang diwakili oleh perwakilan binari iaitu 1011:

$$= 101100 / 1011$$

$$= 1000 \text{ dan baki } 100 \text{ (tiga baki terakhir).}$$

3. Gabungkan 4 bit input yang dimasukkan tadi dengan baki 100. Maka nilai terenkod yang dihasilkan ialah 1001011. Kaedah pengkodan ini juga boleh diimplementasi dengan menggunakan get XOR dan penjadual tukar. (shift register) seperti yang ditunjukkan pada rajah di bawah.



Rajah 2.3 Lakaran kasar litar pengekod CRC

Rajah 2.3 menunjukkan bagaimana proses untuk pengekodan berlaku. Litar ini adalah litar yang dibina berdasarkan persamaan pariti polinomial. Simbol get tersebut berfungsi sebagai suis untuk mengawal pengaliran bit data dan memastikan nilai bit-bit

tersebut dihantar blok demi blok atau rangka demi rangka. R1, R2 dan R3 berfungsi sebagai penjadual tukar (shift register).

Katakan bit maklumat yang ingin dimasukkan adalah 1011. Apabila get dihidupkan, keempat- bit tersebut akan ditambahkan dengan tiga nilai kosong memandangkan jika bit maklumat ini ditukarkan dalam bentuk binari, maka nilainya akan menjadi 1011100. Nilai awalan R1, R2 dan R3 adalah 0. kemudian apabila kesemua nilai bit tersebut telah melalui penjadual, nilai baki akan terbentuk.

Setelah selesai, bit maklumat dan bit baki tadi akan dicantumkan membentuk 7 bit maklumat. Get akan dimatikan dan data 7 bit akan dihantar ke saluran untuk dihantar ke komputer penerima. Penerangan mengenai pengaliran data melalui litar dengan lebih terperinci akan dibincangkan semasa proses rekabentuk pada fasa empat nanti.

2.4 PERBINCANGAN KOD- KOD LINEAR

Kebanyakan cara penyimpanan data yang menggunakan *SRAM (Static Random Memory Access)* boleh menghasilkan ralat kerana sifat radiasi yang terdapat dalam komponen tersebut. Radiasi tersebut menyebabkan perubahan pada maklumat yang hendak disampaikan. Ini adalah masalah yang biasa terjadi dalam penghantaran data melalui satelit dimana kod Hamming tidak berupaya untuk menangani ralat yang terhasil semasa proses penghantaran maklumat jenis ini.

Aplikasi kaedah Hamming ini digunakan di dalam ingatan, sistem bas, unit pemprosesan data dan di *RAM* untuk tujuan pengesanan dua bit ralat dan pembetulan satu bit ralat.

Selain daripada itu terdapat juga inisiatif untuk memperbaiki kod Hamming dalam ingatan. Sebagai contoh terdapat cadangan untuk menambahkan nilai ketumpatan

dalam cip yang digunakan untuk proses pengekodan dan penyahkodan ini. Nilai vot (*voters*) diperlukan untuk menentukan ralat mana yang dikehendaki, memerlukan ruang untuk lewahan secara statik atau dinamik. Perkakasan yang ditambah adalah seperti “memory voters” dan pengesan ralat.

Ulasan telah dibuat oleh Prof. Dr. Mirosław Malik, dari Universiti Zu Berlin untuk meningkatkan tahap pengesanan ralat kod Hamming ini sehingga ke 32 bit. Rekabentuk ini diimplementasikan dalam ingatan komputer dan dinamakan rekabentuk SEC-DED. Sistem ingatan ini akan berupaya untuk membetulkan semua ralat bit tunggal, mengesan semua ralat dua bit dan kadang kala ia berupaya untuk mengesan ralat tiga bit. Masa yang diambil juga sangat pantas iaitu pengesanan ralat mengambil masa 32 ns, dan pembetulan ralat mengambil masa 64 ns. Rekabentuk SEC-DED ini menggunakan dua kitar baca dan tulis. Di mana sekiranya terjadi ralat pada kitar baca dan tulis yang pertama, misalnya terdapat ralat pada bit pariti yang pertama, maka nilai data yang akan dihantar akan mempunyai nilai pelengkap untuk bit pariti yang ralat tersebut.

Pada kitar yang kedua, operasi baca dan tulis akan dijalankan sekali lagi, dimana nilai data yang dihantar semasa kitar kedua akan dikesan sekali lagi sekiranya ia mempunyai ralat. Jika tidak nilai bit pelengkap pariti yang terdapat dalam data tadi akan diterbalikkan dan nilai bit yang salah akan dibetulkan. Ini membenarkan pengesanan dilakukan dengan baik dan dapat mengesan sekiranya ralat terjadi. Ia berbeza dengan kaedah Hamming terdahulu, dimana ia membaca data yang dihantar sekali sahaja. Oleh kerana kaedah rekabentuk ingatan SEC-DED mempunyai nilai pengesanan ralat yang sesuai dengan teknologi terkini, maka ia banyak digunakan dalam aplikasi

pengkomputeran, contohnya sistem UNIVAX 11000/60 yang berupaya mengembalikan isyarat kepada peranti yang membuat permintaan sekiranya ralat dikesan.

Untuk CRC pula, ia biasanya digunakan untuk komunikasi satelit iaitu ia digunakan untuk protokol HDLC. Begitu juga dengan kod “convolutional” yang juga digunakan untuk komunikasi satelit. Kajian juga telah dibuat untuk mencari perbezaan antara pengekodan “convolutional codes” dengan kod blok. Kaedah Hamming dan pemeriksaan lewahan berkisar termasuk dalam kod blok.

Kod linear tidak bergantung kepada nilai sebelumnya yang telah dikodkan, tetapi “convolutional codes” bergantung kepada unit masa dan juga input sebelumnya. Kod linear tidak melibatkan penggunaan memori semasa operasinya tetapi “convolutional codes” melibatkan turutan memori.

Kaedah “convolutional codes” ini menggunakan multipleksor penambah untuk menselarikan nilai antara hasil output yang sama dan hasil output selepasnya. Litar untuk membina kod ini juga adalah lebih kompleks daripada litar hamming dan litar pemeriksaan lewahan berkisar. Oleh kerana “convolutional codes” ini menggunakan nilai yang sebelumnya untuk mendapatkan hasil akhirnya, maka jika terdapat ralat pada output terenkod yang sebelumnya, output yang selepasnya juga akan dikesan ralat walaupun tidak mempunyai ralat.

Kaedah “convolutional codes” adalah amat kompleks berbanding kod linear. Kod ini digunakan untuk komunikasi satelit sama aplikasinya dengan kod pemeriksaan lewahan berkisar sementara kod hamming diaplikasikan dalam ingatan.

Oleh kerana kod “convolutional codes” adalah sangat kompleks, maka untuk diaplikasikan bagi penghantaran data ke satelit ia menggunakan lebar jalur dua kali lebih banyak daripada kod biasa.

Sebagai kesimpulannya, kaedah pengkodan Hamming ini adalah amat ringkas dan perlu diperbaiki untuk menghasilkan kaedah pengesanan ralat. Kaedah lewahan bit berkitar pula adalah amat berkuasa sehingga digunakan untuk penghantaran data untuk aplikasi Ethernet dan Internet dan praktikal berbanding “convolutional codes” kerana mudah dilaksanakan dengan kos yang rendah.

BAB TIGA

METODOLOGI

3.0 METODOLOGI

Dalam bab ini perkara yang akan dibincangkan ialah keperluan perkakasan dan perisian yang digunakan dalam kajian ilmiah ini.

3.1 VHDL

VHDL atau "*Very High Integrated Circuit Hardware Description Language*" adalah bahasa pengaturcaraan untuk penmodalan peranti perkakasan dalam bentuk sistem digital.

Salah satu kelebihan VHDL adalah ia merupakan standard yang digunakan dalam industri elektronik. VHDL telah ditetapkan sebagai standard IEEE 1076-1, *Standard for VHDL*, pada bulan Disember 1987. Menurut IEEE 1076-1, "VHDL adalah bahasa yang digunakan untuk menggambarkan struktur, tingkahlaku, dan aliran data dalam peranti elektronik."

BAB TIGA

Rekabentuk Kefungsian

Rekabentuk RTL

METODOLOGI

Rekabentuk Litar

Rajah 3.0: Rajah pengaliran proses dalam VHDL

3.0 METODOLOGI

Dalam bab ini perkara yang akan dibincangkan ialah keperluan perkakasan dan perisian yang digunakan dalam kajian ilmiah ini.

3.1 VHDL

VHDL atau “*Very High Integrated Circuit Hardware Description Language*” adalah bahasa pengaturcaraan untuk permodelan peranti perkakasan dalam bentuk sistem digital. Ia dijalankan di dalam perisian WEB PACK 4.20 yang digunakan. VHDL telah disahkan oleh American Institute of Electrical and Electronics Engineers pada bulan Disember 1987 dengan piawai IEEE 1076.1.

Keperluan



Rekabentuk Kefungsian



Rekabentuk RTL



Rekabentuk logik



Rekabentuk litar

Rajah 3.0 Rajah pengaliran proses dalam VHDL

Carta alir di atas menunjukkan carta alir aktiviti untuk rekabentuk digital. Peringkat pertama adalah peringkat keperluan. Pada peringkat ini, semua spesifikasi keperluan akan disenaraikan. Keperluan untuk prestasi akan dapat ditentukan melalui gambaran imej yang akan diproses dan operasi yang akan dijalankan ke atas komponen tersebut. Peringkat rekabentuk yang dijalankan dalam projek ini adalah secara “bottom up”. Modul- modul akan dibangunkan dan akan diuji terlebih dahulu sebelum digabungkan kesemuanya untuk membentuk sistem.

Peringkat kedua akan adalah peringkat rekabentuk kefungsian. Keperluan kefungsian ini akan dijalankan pada aras kelakuan (behavioral level). Pada aras ini tiada lagi kitar jam atau isyarat yang akan dipindahkan. Pengkodan dan simulasi juga dijalankan pada peringkat ini. Simulasi adalah untuk mengesahkan kefungsian untuk setiap komponen yang terdapat dalam rekabentuk. Sekiranya terdapat ralat, pembetulan akan dilakukan pada peringkat pengkodan.

Pada peringkat ketiga, rekabentuk dibahagikan kepada elemen-elemen penstoran dan kombinasi logik. Peringkat ini dinamakan “*Register Transfer Level*”. Perincian terhadap rekabentuk akan dilakukan pada aras jadual (registers), ingatan, arithmetik dan status mesin. Perincian RTL ini akan menghasilkan rekabentuk logik yang akan mengimplementasikan setiap komponen dalam rekabentuk tersebut. Tujuan peringkat ini adalah untuk memastikan konfigurasi memenuhi spesifikasi pada aras get iaitu apabila semua komponen dalam rekabentuk ini disepadukan.

Hasil daripada perincian RTL, rekabentuk logik akan dapat dihasilkan. Kesalahan dari peringkat simulasi akan menyebabkan ralat pada aras pembentukan logik.

Terdapat Peringkat yang kelima adalah peringkat rekabentuk litar dimana simulasi pemasaan dan analisis litar akan dilakukan.

Terdapat beberapa komponen yang terdapat dalam VHDL iaitu:-

- Bahasa sebenar VHDL
- Deklarasi untuk jenis data (pakej “STANDARD”)
- Fungsi utiliti (pakej “TEXTIO” – teks input output)
- Rekabentuk pengguna (pakej “WORK”)
- Perpustakaan STD atau STD Library termasuk pakej “STANDARD” dan pakej “TEXTIO” iaitu:-

- Pakej vendor
- Perpustakaan vendor
- Pakej pengguna
- Perpustakaan pengguna

Dalam pembangunan projek ini, bahasa pengaturcaraan VHDL ini akan digunakan bersama teknologi FPGA. FPGA adalah “*Field Programmable Arrays*”. Teknologi ini bermakna persamaan Boolean akan digunakan semasa pengaturcaraan. Persamaan Boolean ini akan dibahagikan kepada sub-sub unit untuk dikonfigurasi kepada blok logik untuk FPGA atau “*Configurable logic blocks*”, (CLB).

Selain daripada teknologi FPGA, terdapat juga kaedah yang dikenali sebagai ASICS atau “*Application Specifics Integrated Circuits*”. Kaedah ini dijalankan dengan menggunakan kaedah pemindahan automatik daripada kod – kod VHDL kepada aras get.

Terdapat beberapa kelebihan VHDL iaitu:-

- Dengan melaksanakan pengaturcaraan jenis ini, ia seperti melaksanakan kod yang sebenar.
- Boleh memodelkan kelakuan sistem (*system behavioral*) daripada secara terus daripada teknologi yang dipilih. Contohnya boleh memodelkan pengkod-penyahkod.
- Oleh kerana VHDL telah mempunyai piawainya, maka sebarang pembangunan produk yang menggunakan piawai tersebut akan bertahan lama tanpa mengalami sebarang ancaman jika tidak serasi (*incompatible*) dengan perisian yang lain.
- Ia juga mendapat sokongan daripada kerajaan Amerika. Maka penggunaannya meluas untuk komponen elektronik.
- Pengaturcaraannya juga disokong oleh kebanyakan industri untuk kerana nilai komersialnya.
- Pengaturcaraannya mudah disimulasikan dengan pada pelbagai jenis komponen, peralatan dan pada pelbagai aras semasa proses pembangunan
- Mempunyai kebolehan memodelkan semua aras rekabentuk. Daripada kotak elektronik hinggalah ke transistor. Memuatkan kelakuan komponen yang diadaptasikan daripada rumus matematik.
- Oleh kerana pengaturcaraan dilakukan dengan menggunakan pendekatan ke atas perkakasan, maka penggunaan unit kawalan berpusat (CPU) tidak akan diganggu.

3.2 WEB PACK 4.2

Perisian yang digunakan untuk menjana kod-kod VHDL adalah perisian Xilinx WEB PACK 4.2. Terdapat beberapa modul yang terdapat di dalam perisian bergantung kepada jenis kerja yang hendak dibuat iaitu:-

1. Design entry

“design entry and synthesis tools” adalah bahagian utama dalam perisian ini. Ia digunakan untuk hampir keseluruhan proses. Memilih salah satu modul keluarga iaitu Spartan, Virtex atau CPLD untuk menggunakan perisian dengan mudah. Antara kandungan yang terdapat dalam modul “Design Entry ” ternasuk:-

- Skematik ECS untuk rekabentuk aras tertinggi (top level design)berasaskan berasaskan HDL
- Sintesis XST untuk menyokong VHDL dan rekabentuk verilog
- Project Navigator, antaramuka yang terdapat di dalam WEB PACK 4.2
- Simulasi VHDL dan Verilog.

2. CPLD Fitter

Ia menyesuaikan pemuatan peranti dan perisian pengujian untuk merekabentuk mana-mana rekabentuk Xilinx yang berasaskan keluarga CPLD seperti Xilinx

XC9500, XC9500XL, XC9500XV dan sebagainya. Modul ini boleh digunakan untuk melaksanakan rekabentuk dari dua sumber iaitu:-

4. Perlaksanaan Virtex

- Rekabentuk menggunakan bahasa VHDL, Verilog atau ABEL yang disokong modul "Design Entry".
- Netlist EDIF atau XNF yang disediakan menyediakan peralatan sokongan pihak ketiga "Alliance Eda Tool".

Komponen yang terdapat dalam CPLD Fitter ialah:-

- Penganalisa masa
- Perisian pemuat CPLD (cpldfit command yang menyokong semua keluarga CPLD)
- Model penjana simulasi masa untuk CPLD untuk menghasilkan model VHDL atau Verilog
- Penyunting kekangan

3. Perlaksanaan Spartan

Menyediakan semua perlaksanaan peranti dan pengujian perisian yang diperlukan untuk memetakan rekabentuk ke dalam keluarga "SpartanII" atau Spartan-IIE.

Perlaksanaan Spartan termasuk:-

- Model yang terdapat dalam CPLD Programmer termasuk -
- Model simulasi pemasaan untuk FPGA untuk menghasilkan model VHDL atau Verilog.
- Penyunting kekangan.

- Penganalisa pemasaan.

4. *Perlaksanaan Virtex*

Menyediakan implementasi dan perisian pengujian yang diperlukan untuk memetakan rekabentuk kepada peranti Virtex – E atau virtex- II. Perlaksanaan Virtex ini boleh digunakan untuk semua program yang menggunakan peranti FPGA dan Serial PROM diperolehi dari dua sumber iaitu:-

- Rekabentuk menggunakan bahasa VHDL, Verilog atau ABEL yang disokong perisian pengaturcaraan Xilinx impact modul “*Design Entry*” .
- Netlist EDIF atau XNF yang disediakan menyediakan peralatan sokongan pihak ketiga “Alliance Eda Tool”.

Implementasi Virtex termasuk:-

- Model simulasi pemasaan untuk FPGA untuk menghasilkan model VHDL atau

Verilog

- Penyunting kekangan
- Penganalisa pemasaan

5. *CPLD Programmer*

Digunakan untuk keluarga CPLD seperti Xilinx XC9500, X9500XL dan sebagainya.

- Modul yang terdapat dalam CPLD Programmer termasuk:-
- Perisian pengaturcaraan Xilinx impact
- Fail BSDL untuk peranti CPLD

6. Pengaturcara FPGA

digunakan untuk semua program yang menggunakan peranti FPGA dan Serial Proms.

- Modul yang terdapat dalam FPGA Programmer adalah:-

perisian pengaturcaraan Xilinx impact.

- Penformat fail PROM

- Fail BSDL untuk semua peranti Xilinx dan Serial PROM

7. StateCAD

Peralatan Xilinx StateCAD akan menjana rekabentuk VHDL, Verilog dan ABEL FSM.

8. HDL Benchner

Peralatan Xilinx HDL Benchner menjana "test benches" untuk VHDL dan Verilog. Ia membantu merekabentuk dan menyunting stimulasi bentuk gelombang fizikal dan akan dieksport ke dalam "test bench" untuk VHDL dan Verilog untuk dimasukkan dalam simulasi Project Navigator.

9. ChipViewer

Ia adalah antaramuka grafik untuk melihat kesesuaian pin dan logic dan untuk masuk ke lokasi pin I/O dan untuk melihat semua perkara yang terdapat dalam CPLD untuk Xilinx.

10. *CPLD Schematic Capture Libraries*

Dalam perpustakaan komponen skematiknya menyediakan simbol termasuk rekabentuk skematik ECS yang disediakan menyediakam modul “Design Entry”.

11. *FPGA Schematic Capture Libraries*

Dalam perpustakaan komponen skematiknya menyediakan simbol termasuk rekabentuk skematik ECS yang disediakan menyediakam modul “Design Entry”.

12. *XPower*

Ia adalah peralatan untuk menganalisis yang menggunakan data rekabentuk dan peranti .Maklumat dipersembahkan dalam bentuk grafik dan format laporan ASCII . Ia juga menyokong Cool Runner XPLA3 dari keluarga CPLD dan Virtex II FPGA.

13. *Peranti sokongan*

WebPACK menyokong perlaksanaan untuk keluarga Xilinx yang berikut:-

- XC9500XL (3.3V CPLDs)
- XC9500XV (2.5V CPLDs)
- XCR3000XL (CoolRunner XPLA3 3.3V zero-standby CPLDs)
- XC2C00 (CoolRunner-II 1.8V zero-standby CPLDs)
- XC2S00 (Spartan-II FPGAs)
- XC2S00E (Spartan-II E FPGAs)
- XCV00E (Virtex-E FPGAs up to XCV300E)
- XC2V00 (Virtex-II FPGAs up to XC2V250)

- XC9500 (5V CPLDs)

3.3 KEPERLUAN SISTEM

Sistem perisian :

- Windows NT 4.0 (Service Pack 5 atau ke atas)
- Windows 98 (original and SE)
- Windows 2000
- Windows Millennium

4.2 ALIRAN REKABENTUK

Dalam bab ini, perikar yang akan dikembangkan adalah sub-unit modul yang terdapat di dalam rekabentuk luar yang disediakan, merangkumi diagram aras tertinggi (top level diagram) untuk keseluruhan sistem, diagram blok, perancangan aliran proses merangkumi kumpulan-kumpulan yang disediakan. Aliran rekabentuk yang dikembangkan ialah "bottom up", iaitu ialah kerangka modul pengendali akan dihubungkan bersama modul perantara.

4.1 DIAGRAM ARAS

BAB EMPAT REKABENTUK



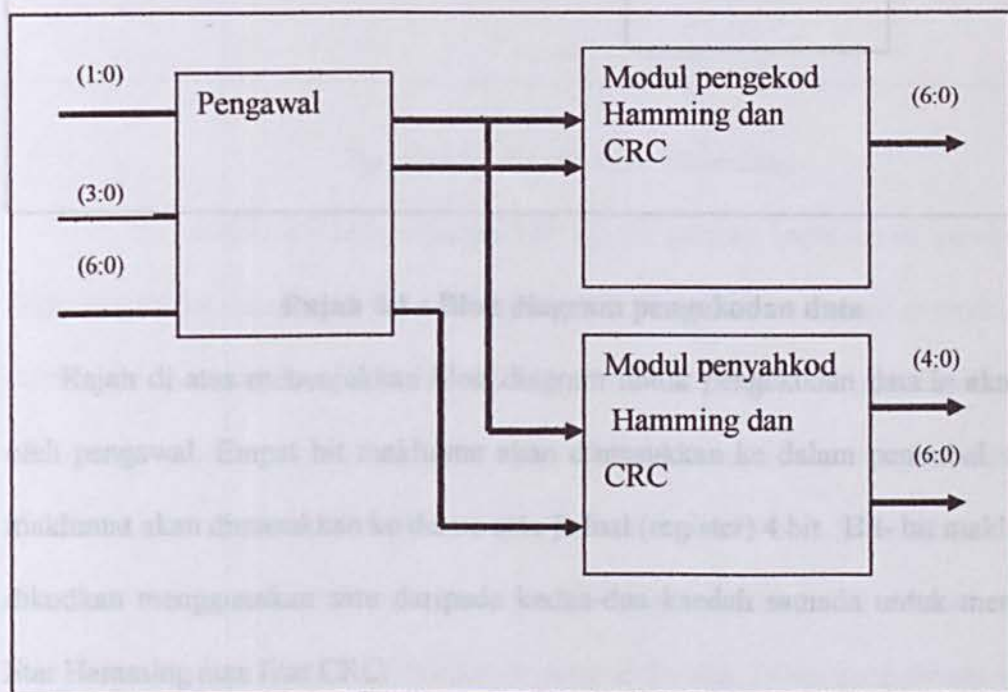
Fig. 4.1 Top Level Diagram Aras Tertinggi

Fig. 4.1 menunjukkan diagram aras tertinggi untuk keseluruhan rekabentuk sistem. Modul perantara terdiri daripada 7 blok input dan 7 keluaran output. Modul pengendali terdiri daripada 7 blok input dan 7 keluaran output. Diagram ini menunjukkan aliran data antara modul-modul tersebut.

4.0 ALIRAN REKABENTUK

Dalam bab ini , perkara yang akan dibincangkan adalah sub-sub modul yang terdapat di dalam rekabentuk litar yang dicadangkan, merangkumi diagram aras tertinggi (top level diagram) untuk keseluruhan sistem, diagram blok, penerangan aliran proses merangkumi komponen –komponen yang dicadangkan. Aliran rekabentuk yang dicadangkan ialah “bottom up”. Ini adalah kerana modul pengekod akan digabungkan bersama modul penyahkod.

4.1 DIAGRAM ARAS TERTINGGI (TOP LEVEL DIAGRAM)

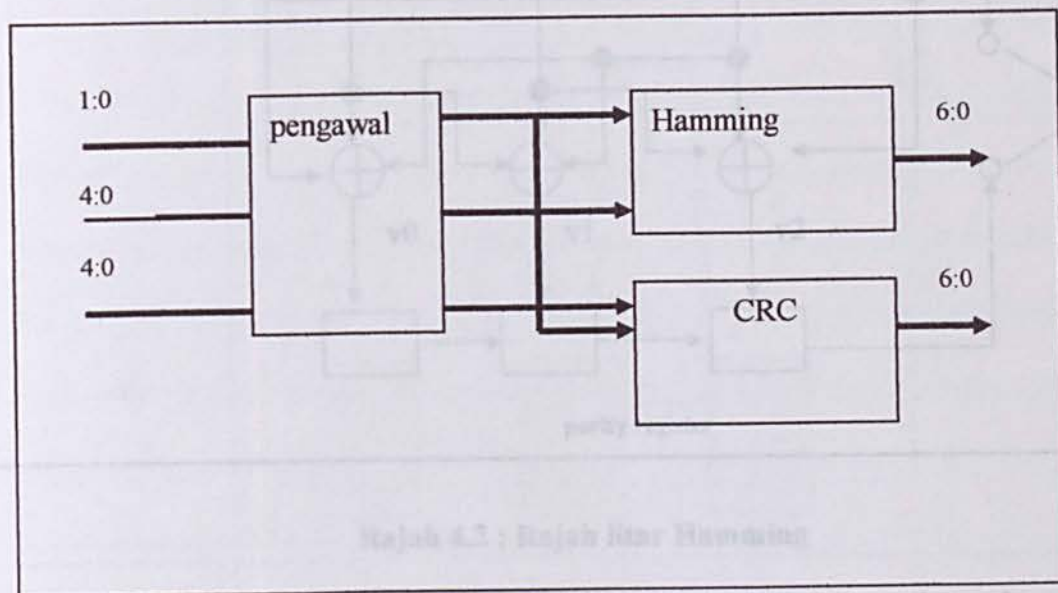


Rajah 4.0 Rajah Diagram Aras Tertinggi

Rajah di atas menunjukkan diagram aras tertinggi untuk keseluruhan rekabentuk sistem . Modul pengkodan mempunyai 4 masukan input dan 7 keluaran output. Modul ini terdiri daripada pengkod Hamming dan CRC. Pengawal terdiri daripada operator untuk membolehkan pengguna memilih kaedah pengkodan data yang hendak

digunakan Begitu juga dengan modul penyahkodan yang dibangunkan secara berasingan dalam projek ini.

4.2 DIAGRAM ARAS TERTINGGGI MODUL PENGEKOD

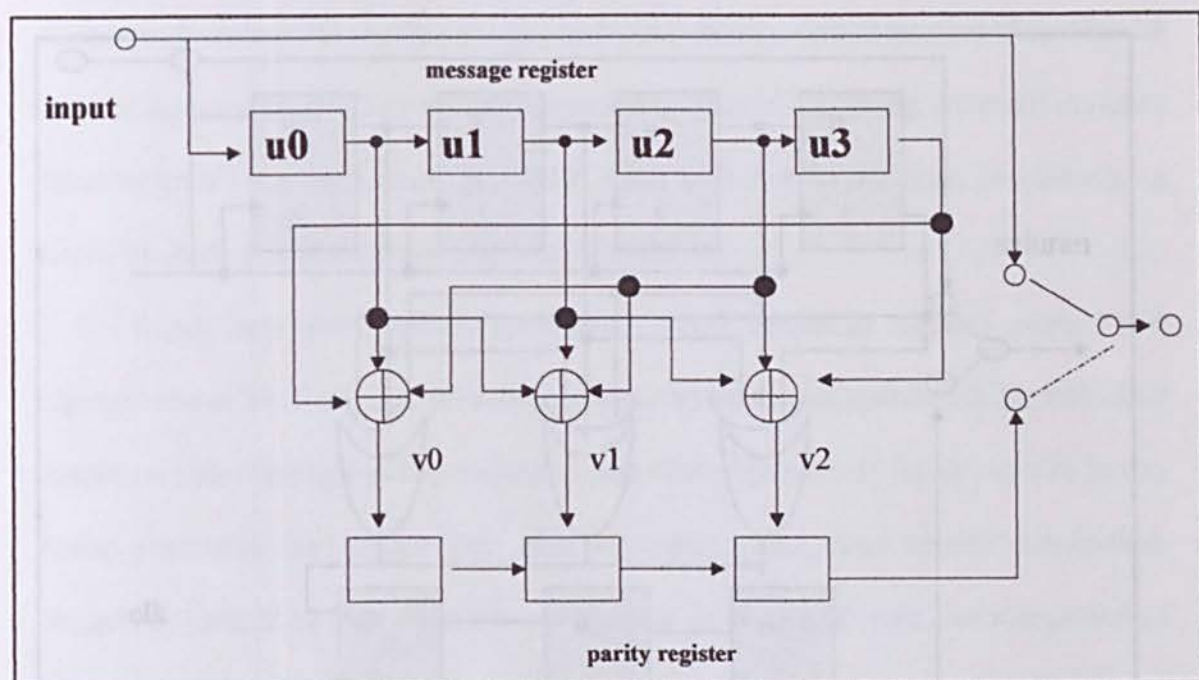


Rajah 4.1 : Blok diagram pengkodan data

Rajah di atas menunjukkan blok diagram untuk pengkodan data. Ia akan dikawal oleh pengawal. Empat bit maklumat akan dimasukkan ke dalam penimbal. Semua bit maklumat akan dimasukkan ke dalam satu jadual (register) 4 bit. Bit-bit maklumat akan dikodkan menggunakan satu daripada kedua-dua kaedah samada untuk menggunakan litar Hamming atau litar CRC.

Pengguna akan masukkan nilai mod pilihan samada 01 atau 10. Jika nilai 01 dimasukkan, maka kaedah pengkodan Hamming akan dipilih, dan jika pengguna memasukkan nilai 10, maka litar CRC akan digunakan. Kemudian tujuh bit data mesej akan dikeluarkan dan hasilnya akan dihantar ke penyahkod pada bahagian penerima.

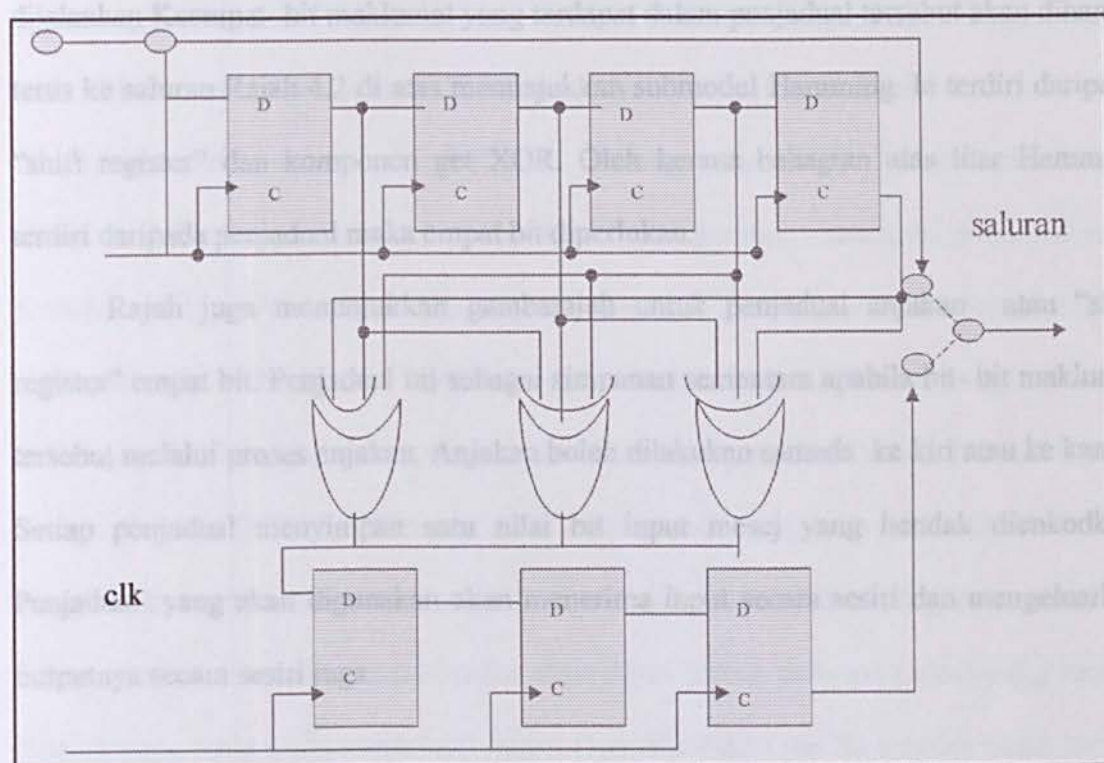
4.3 PENGEKODAN HAMMING



Rajah 4.2 : Rajah litar Hamming

Rajah 4.2 yang ditunjukkan adalah lakaran kasar untuk pembinaan litar Hamming. Nilai input data yang dimasukkan adalah 1011. Nilai v_0 mewakili nilai bit pariti pertama. Input dari u_0 , u_3 dan u_2 akan di“XOR” kan untuk mendapatkan nilai bit pariti yang pertama. Kemudian input dari u_0 , u_1 dan u_3 akan menghasilkan nilai v_1 . Manakala input dari u_1 , u_2 dan u_4 akan menghasilkan nilai bit v_3 . Kesemua nilai v_1 , v_2 dan v_3 akan dimasukkan ke dalam penjadual masing-masing dan dihantar ke saluran untuk digabungkan dengan bit maklumat yang sedia ada. Proses penggabungan ini akan membentuk data yang telah terenkod.

4.3.1 OPERASIAN KOD HAMMING



Rajah 4.3 : Litar Pengekod Hamming

Rajah 4.2 yang ditunjukkan sebelum ini adalah setara dengan rajah 4.3 di atas.

Bit yang pertama akan memasuki litar dan diletakkan ke dalam penjadual yang terkiri sekali. Kemudian apabila bit kedua masuk ke dalam litar, bit yang pertama akan dialihkan ke storan yang kedua iaitu ke kanan. Kemasukan bit ke dalam litar adalah secara sesiri. Begitu juga seterusnya sehingga bit yang terakhir. Ini adalah kerana litar ini dikawal oleh "clock" yang berfungsi untuk mengawal pemasaan dan nilai input masuk mengikut detiknya. "Shift register" ini mempunyai empat detik "clock" dianggap sebagai t_1, t_2, t_3 dan t_4 . Selagi nilai "clock" adalah 1, maka kemasukan input secara selari adalah dibenarkan. Setelah sampai ke t_4 , maka status "clock" akan kembali kepada 0 dan tiada lagi kemasukan input.

Nilai Gambarajah 4.4 di sebelah menunjukkan pengekodan CRC menggunakan 4 bit data menggunakan kaedah pemeriksaan lewahan berkitar. Tiga penjadual dengan nilai 0 perlu diletakkan ke dalam litar. Bit pertama yang akan dimasukkan ialah 1. Ia akan dimasukkan dari kawasan bertanda "data". Nilai g yang berada pada bahagian tengah litar mewakili nilai polinomial penjana. Apabila bit pertama memasuki penjadual yang pertama sekali, maka ia akan digabungkan dengan bit 1 yang sedia ada untuk nilai g yang pertama. Kemudian $1 \text{ "AND"} 1$ akan menghasilkan 1. Nilai 0 yang sedia ada dalam penjadual yang pertama akan dikeluarkan dari penjadual tersebut. Nilai 0 yang keluar akan di "XOR"kan dengan nilai 1 yang sedia ada pada g2. $1 \text{ XOR } 0$ akan menghasilkan nilai 1. Nilai 1 ini akan dimasukkan ke dalam jadual yang kedua dan nilai 0 yang sedia ada dalam jadual kedua akan dipindahkan pada penjadual yang ketiga. Nilai 0 yang lama pada penjadual yang ketiga akan dihantar ke saluran untuk tujuan pembentukan baki polinomial. Penjadual yang ketiga terletak dibahagian terkanan litar. Maka untuk anjakan yang pertama nilai dalam ketiga-tiga penjadual ialah 110 dan nilai yang berada dalam saluran ialah 0.

Apabila anjakan kedua berlaku, nilai input yang kedua dimasukkan. Nilai kandungan dalam penjadual ketiga akan di "XOR" kan dengan nilai input. $1 \text{ XOR } 0$ akan menghasilkan 1. Nilai bit yang pertama akan dialihkan ke penjadual kedua iaitu ke kanan. Nilai penjadual untuk anjakan yang pertama ialah 110. Apabila nilai input kedua iaitu bit 1 dimasukkan, maka nilai tersebut akan menggantikan nilai 1 yang sedia ada dalam penjadual terkiri. Nilai 1 yang lama ini akan dikeluarkan dari penjadual dan di "XOR" kan dengan nilai 1 yang sedia ada pada g2. Nilai 0 akan dihasilkan dan diletakkan dalam penjadual kedua menggantikan nilai 1 yang sedia ada. Nilai 1 dalam penjadual kedua tersebut akan dialihkan dan diletakkan dalam penjadual yang ketiga.

Nilai 0 pada penjadual ketiga akan dihantar ke saluran untuk tujuan pembentukan baki polinomial. Maka untuk “anjakan” yang kedua akan menghasilkan nilai 101 dalam ketiga-tiga penjadual tersebut dan nilai yang berada dalam saluran ialah 00.

Pada anjakan yang ketiga, nilai 0 akan dimasukkan. Nilai terdahulu dalam ketiga-tiga penjadual tersebut adalah 101. Nilai 1 dalam penjadual yang terkanan sekali akan di “XOR”kan dengan nilai 0 tersebut. Nilai 1 akan dihasilkan. Nilai 1 akan dimasukkan ke dalam penjadual yang terkiri sekali menggantikan nilai 1 yang sedia ada. Nilai 1 yang sedia ada akan dikeluarkan dari penjadual yang pertama dan di “XOR” kan dengan dengan nilai g_2 yang sedia ada. $1 \text{ XOR } 0$ akan menghasilkan nilai 0. Nilai 0 dari penjadual kedua akan dikeluarkan dan diletakkan ke dalam penjadual yang terkanan sekali yang berada di dalam litar. Nilai 1 pada penjadual yang ketiga akan dihantar ke saluran untuk tujuan pembentukan baki polinomial. Maka nilai yang berada dalam saluran sekarang ialah 100.

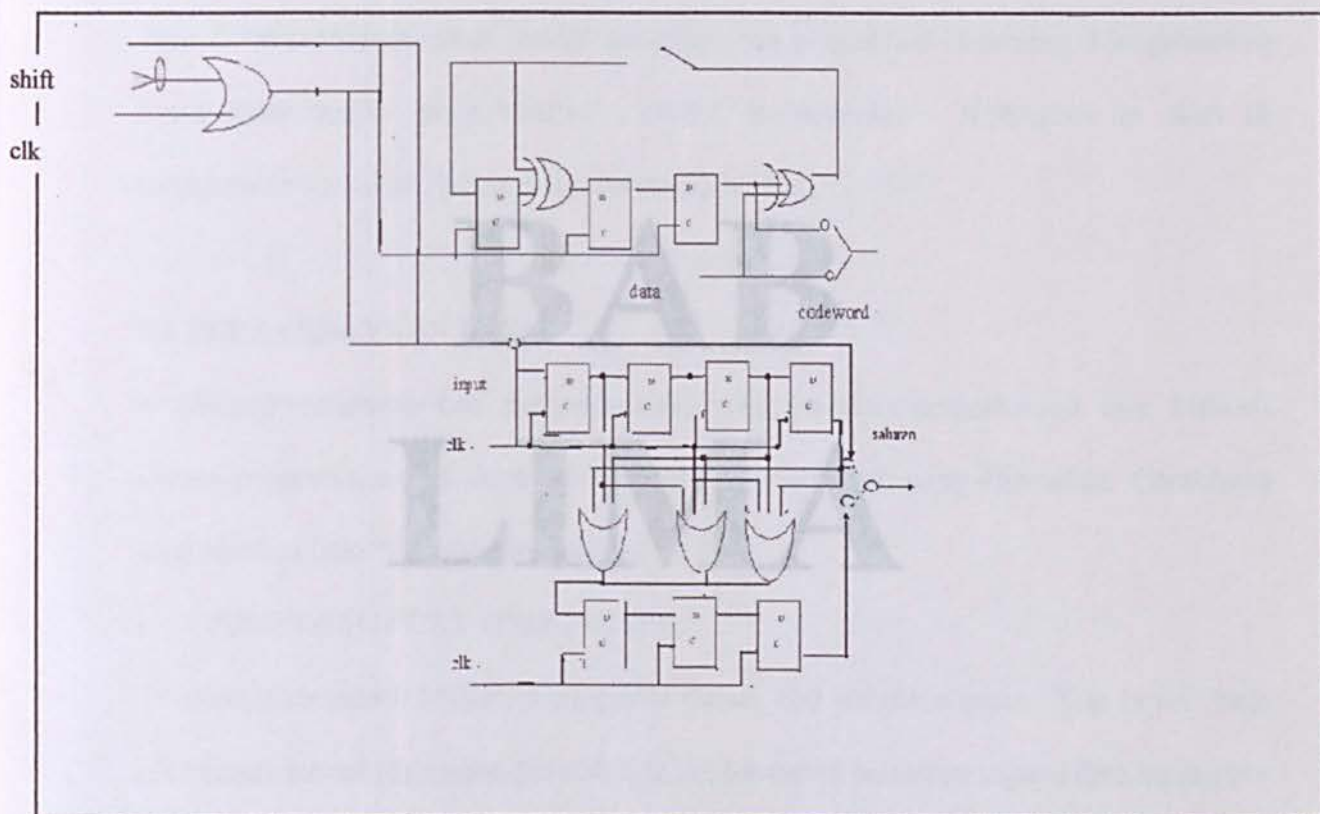
Proses yang sama akan berlaku dalam anjakan yang keempat menyebabkan nilai yang terakhir pada penjadual ialah 100. Maka nilai 100 yang terbentuk akan digabungkan dengan nilai bit data iaitu 1011 akan digabungkan membentuk “codeword” 1001011.

Rejoh 4.3 : Litar gabungan pengkod

Pemrograman yang terperinci tentang litar Hamming dan pemeriksaan kesilapan berkait telah dinyatakan sebelum ini. Pemamfaatan untuk litar ini ialah komponen pengawal yang terdapat dalam litar untuk membenarkan pengguna memasukkan input supaya pengguna boleh memilih untuk menggunakan mana-mana kaedah pengkodan. Rejabentuk pada perintang ini tidak akan diwujudkan kerana perintang ini adalah perintang analisis yang akan membekalkan kod pengawarasan yang dibina ke atas cip FPGA dan dipap.

4.5 LITAR GABUNGAN

Oleh kerana perincian pengekodan litar Hamming dan litar CRC, maka akan ditunjukkan dimana kedua-dua litar tersebut akan dicantumkan pada pengawal.



Rajah 4.5 : Litar gabungan pengekod

Penerangan yang terperinci tentang litar Hamming dan pemeriksaan lewahan berkisar telah dinyatakan sebelum ini. Penambahan untuk litar ini ialah komponen pengawal yang terdapat dalam litar untuk membenarkan pengguna masukkan input supaya pengguna boleh memilih untuk menggunakan mana-mana kaedah pengekodan. Rekabentuk pada peringkat ini tidak akan dibangunkan kerana peringkat ini adalah peringkat sintesis yang akan memetakan kod pengaturcaraan yang dibina ke atas cip FPGA dan diuji.

5.0 PERLAKSANAAN

Bagian ini terbagi menjadi 2 fase utama yaitu:-

Fase 1: pemikaran modul Hamming dan modul CRC ke dalam bentuk pengalokasian yang bersamaan.

Fase 2: Perakuturan antara modul pengkode dan parakode Hamming. Mengeluarkan nama-nama output menggunakan modul multipleksor. Kemudian ia akan di terjemahkan ke dalam bahasa pengalokasian VHDL.

BAB LIMA

5.1 PERLAKSANAAN

Secara umumnya kod pengalokasian menggunakan perpustakaan IEEE 1164.all kerana perpustakaan ini mengandungi definisi data yang digunakan. Contohnya nilai residual ialah "0".

1 : PEMETAAN PORT (PORT MAP)

Kod pemetaan ini boleh dijumpai di dalam kod pengalokasian "Top Level" bagi rekabentuk modul Hamming dan CRC. Ia adalah untuk bertujuan supaya data input yang dimasukkan akan melalui semua langkah pengalokasian pada komponen yang ingi diuji dan menghasilkan output yang sepatutnya.

PERLAKSANAAN

II : CLK (UNTUK KIRAH MASA)

Terlepas pabrakan antara peralokasian modul Hamming dengan modul CRC, ia kerana untuk Hamming, data dimasukkan secara serentak (concurrent), sementara data untuk CRC dimasukkan secara berurutan (sequential). Maka pembolehubah "clk", tidak diperlukan dalam pengalokasian Hamming kerana semua data dimasukkan pada masa yang sama dalam 4 pangsaf 4 bit. Sedangkan clk ini akan digunakan untuk

5.0 PERLAKSANAAN

Bahagian ini terbahagi kepada 2 fasa utama iaitu:-

Fasa 1: penukaran modul Hamming dan modul CRC ke dalam bentuk pengaturcaraan yang berasingan.

Fasa 2: Percantuman antara modul pengekod dan penyahkod Hamming. Mengeluarkan mana-mana output menggunakan modul multipleksor. Kemudian ia akan di terjemahkan ke dalam bahasa pengaturcaraan VHDL.

5.1 PERLAKSANAAN FASA 1

Secara umumnya kod pengaturcaraan menggunakan perpustakaan ieee 1164.all kerana perpustakaannya menyokong kod pengaturcaraan yang digunakan. Contohnya nilai resolusi iaitu "Z" dan sebagainya.

i : PEMETAAN PORT (PORT MAP)

Kod pemetaan ini boleh dijumpai di dalam kod pengaturcaraan "Top Level" bagi rekabentuk modul Hamming dan CRC. Ia adalah untuk bertujuan supaya data input yang dimasukkan akan melalui semua langkah pengaturcaraan pada komponen yang ingin diuji dan menghasilkan output yang sepatutnya.

ii : CLK (UNTUK KITAR MASA)

Terdapat perbezaan antara pelaksanaan modul Hamming dengan modul CRC, ini kerana untuk Hamming, data dimasukkan adalah secara segerak (concurrent), sementara data untuk CRC dimasukkan secara berjujukan (sequential). Maka pembolehubah "clk", tidak diperlukan dalam pengekodan Hamming, kerana semua data diumpukkan pada masa yang sama dalam 4 penjadual 4 bit. Sebaliknya clk ini akan digunakan untuk

pengekodan CRC kerana 4 bit data input akan diambil satu persatu untuk langkah yang seterusnya.

5.1.1 TEKNIK PENGEKODAN HAMMING

entity hamenc is

use work.all;

port(datain : in std_ulogic_vector(3 downto 0); --d0 d1 d2 d3

hamout : out std_ulogic_vector (6 downto 0));

end hamenc;

architecture ver2 of hamenc is

signal p0,p1, p2 : std_ulogic; --bit pariti

begin

--menjana bit pariti

p0 <= (datain(0) xor datain(1)) xor datain(3); --1 3+5+7

p1 <= (datain(3) xor datain(2)) xor datain(1); ----2 3+6+7

p2 <= (datain(2) xor datain(1)) xor datain(0); --4 5+6+7

--sambungkan dua output

hamout(6 downto 4) <= (p2,p1,p0);

hamout(3 downto 0) <= datain(3 downto 0);

Dalam teknik pengekodan ini, apa yang penting adalah penjanaan bit- bit pariti. p0 mewakili bit pariti 1, p1 mewakili bit pariti 2 dan seterusnya p2 mewakili bit pariti 4.

Penghasilan 3 bit pariti ini adalah secara segerak dan melalui pelaksanaan get- get XOR, maka kesemua bit-bit pariti ini akan digabungkan dengan 4 data input yang masuk, dan menghasilkan 7 bit output yang telah dienkodkan.

5.1.2 TEKNIK PENGEKODAN PEMERIKSAAN LEWAHAN BERKITAR (CRC)

entity crc_encode is

```
port (
```

```
    Clk,Rst,load    : in std_ulogic;
```

```
    Din             : in std_ulogic_vector(3 downto 0);
```

```
    CRC_Sum         : out std_ulogic_vector(6 downto 0)
```

```
); Sum<= Din & X(2 downto 0);
```

```
end crc_encode;
```

architecture behavior of crc_encode is

```
    signal X          : std_ulogic_vector(6 downto 0);
```

```
    signal s_d        : std_ulogic_vector(3 downto 0);
```

```
begin
```

```
    s_d <= Din;
```

```
    process(Clk,Rst)
```

```
    begin
```

```
        if Rst = '1' then
```

```
            X <= (others => '0'); -- initialize values
```

```
            CRC_Sum <= "00000000";
```

```

    elsif rising_edge(Clk) then
        if (load='1') then
            s_d<='0' & s_d(3 downto 1);
            X(6 downto 3) <= (others=>'0');
            X(2) <= s_d(0) xor X(0);
            X(1) <= s_d(0) xor X(2) xor X(0);
            X(0) <= X(1);
        end if;
    end process;

    CRC_Sum <= Din & X(2 downto 0);

```

X0, X1, dan X2 adalah penimbal untuk menyimpan nilai bit-bit yang masuk secara selari. Din mewakili data input yang masuk. Nilai-nilai Din yang selari akan diambil satu persatu secara selari diumpukkan ke dalam setiap penimbal akan dianjakkan ke kanan, iaitu ke penimbal yang seterusnya, apabila terdapat nilai baru yang masuk ke dalam penimbal tersebut. Bit-bit tersebut akan melalui pelaksanaan get-get XOR dan dianjak sehingga tiba ke penimbal terakhir X2. Kemudian daripada penimbal X2, bit pariti akan dikeluarkan satu persatu sehingga 3 bit selesai dikeluarkan. Keluaran terakhir daripada penimbal X(0) dan akan digabungkan dengan nilai inputnya sebanyak 4 bit .

5.2 FASA 2: PENGGABUNGAN MODUL CRC DAN HAMMING

Modul 1 terdiri daripada penggabungan submodul pengekod dan penyahkod Hamming. Modul 2 pula terdiri daripada penggabungan submodul pengekod dan

penyahkod CRC. Penggabungan sub- sub modul ini akan membenarkan pengguna untuk memilih untuk memasukkan 4 bit data untuk dienkodkan atau memasukkan 7 bit data untuk tujuan penyahkodannya. Satu modul tambahan yang dinamakan sebagai multipleksor akan digunakan untuk menerima output daripada pengekod atau penyahkod Hamming . Struktur multipleksor mengandungi komponen operator atau pengawal yang akan bertindak sebagai pemilih.

5.2.1 MODUL MULTIPLEKSOR HAMMING

entity mux is

```

port (
    operator      : in std_ulogic_vector(1 downto 0);
    host_hamout   : in std_ulogic_vector(6 downto 0);
    host_DOUT     : in std_ulogic_vector(3 downto 0);
    result        : out std_ulogic_vector(6 downto 0)
);

```

end mux;

architecture mux_arch of mux is

```

    signal result1 : std_ulogic_vector(6 downto 0);
    signal result2 : std_ulogic_vector(3 downto 0);
    signal res      : std_ulogic_vector(6 downto 0);

    constant c_enc : std_ulogic_vector(1 downto 0) := "01";
    constant c_dec : std_ulogic_vector(1 downto 0) := "10";

```

begin

result1 <= host_hamout; --when c_dec,

result2 <= host_DOUT; --when c_enc,

with operator select

res <= result1 when c_enc,

result2 & "ZZZ" when c_dec,

"0000000" when others;

result<=res;

end mux_arch;

5.2.1 MODUL 1: PENGABUNGAN PENGKOD DAN PENYAHKOD

Modul multipleksor Hamming akan menerima output daripada proses pengkodan atau proses penyahkodan. Sekiranya proses pengkodan dipilih nilai operator 01 akan dimasukkan ke dalam "testbench". Outputnya akan dimasukkan ke dalam nilai result.

5.2.2 MODUL MULTIPLEKSOR CRC

constant c_enc : std_ulogic_vector(1 downto 0) := "01";

constant c_dec : std_ulogic_vector(1 downto 0) := "10";

begin

CRCencod <= CRC_en;

CRCdecod <= CRC_de & "000000";

with operator select

last <= CRCencod when c_enc,

CRCdecod when c_dec, (5 downto 0) --utk decoder

"0000000" when others;

result <= last;

Cara yang sama juga digunakan untuk CRC. Sekiranya output daripada submodul daripada pengekod dipilih, operator 01 akan dipilih dan nilai CRCencod sebanyak 7 bit akan dipilih. Begitu juga sebaliknya dengan penyahkodnya. 7 bit output ternyahkod akan dikeluarkan, ini adalah kerana CRC hanya boleh mengesan ralat tetapi tidak berupaya untuk membetulkannya.

5.2.3 MODUL 1: PENGGABUNGAN PENGKOD DAN PENYAHKOD HAMMING.

Untuk penggabungan modul ini, komponen- komponennya adalah terdiri daripada kod pengaturcaraan untuk Hamming, penyahkod Hamming, multipleksor, "top level untuk modul Hamming" dan "test bench" untuk untuk "top level" tersebut. Kod pengaturcaraannya adalah seperti yang ditunjukkan dibawah.

entity toplevel is -- black box untuk host

port (operator : in std_ulogic_vector(1 downto 0);

datain : in std_ulogic_vector(3 downto 0); --d0 d1 d2 d3 input utk encoder

DI : in std_ulogic_vector(6 downto 0); --utk decoder

SY : out std_ulogic_vector(2 downto 0);

DO : out std_ulogic_vector(6 downto 0); --7 bit untuk tentukan ralat

DOU : out std_ulogic_vector(6 downto 0);


```

        result : out std_ulogic_vector(6 downto 0) -- utk decoder
    );

end toplevel;

architecture toplevel_arch of toplevel is

    component hamenc -- encoder untuk hamming

    port (

        datain : in std_ulogic_vector(3 downto 0); --d0 d1 d2 d3 input utk encoder

        hamout : out std_ulogic_vector(6 downto 0) );

    end component;

    component HAMDEC --decoder untuk hamming

    port (

        DI : in std_ulogic_vector(6 downto 0);

        SY : out std_ulogic_vector(2 downto 0); --4 bit output

        DO : out std_ulogic_vector(6 downto 0); --7 bit untuk tentukan ralat

        DOU : out std_ulogic_vector(6 downto 0); -- 7 bit betul

        DOUT : out std_ulogic_vector(3 downto 0) );

    end component;

    component mux -- component untuk host//input untuk setiap modul jadi output

    port ( operator : in std_ulogic_vector(1 downto 0);

        host_hamout : in std_ulogic_vector(6 downto 0);-- hammout input databit

        host_DOUT : in std_ulogic_vector(3 downto 0);

```

```

submodule result : out std_ulogic_vector(6 downto 0)
    component U1 : hamenc port map (
        datain => datain,
        hamout => s_result1); -- 0 to 6 output untuk encoder
    component U2 : HAMDEC port map (
        DI => DI, -- 7 bit input
        SY => SY,
        DO => DO,
        DOU => DOU,
        DOUT => s_result2 ); -- 4 bit output
    component U3 : mux port map (
        operator => operator,
        host_hamout => s_result1, --0 to 3
        host_DOUT => s_result2,
        result => result ); -- 0 to 6
end component;

signal s_result1 : std_ulogic_vector(6 downto 0);
signal s_result2 : std_ulogic_vector(3 downto 0);

begin -- portmap untuk input,output, clk utk setiap modul

entity toplvl is
    port (
        DI : in std_ulogic_vector(7 downto 0);
        SY : in std_ulogic_vector(6 downto 0);
        DO : in std_ulogic_vector(3 downto 0);
        DOU : in std_ulogic_vector(4 downto 0);
        operator : in std_ulogic_vector(2 downto 0);
        result : out std_ulogic_vector(6 downto 0);
    );
end entity;

```

Untuk pengaturcaraan “top level” Hamming, sub- submodulnya akan digabungkan sebagai komponen. Pin input adalah terdiri daripada 4 bit masukan input pengekod, 7 bit masukan input penyahkod dan operator. Pin outputnya pula terdiri daripada 7 bit output untuk bit yang telah dikodkan, serta beberapa pin output untuk

submodul penyahkod. Semua data yang dimasukkan dalam “test bench ” akan melalui komponen yang berkenaan dan seterusnya akan dipetakan nilai- nilai masukan tersebut pada rekabentuk “top level”.

5.2.4 MODUL 2: PENGGABUNGAN MODUL PENGKOD DAN PENYAHKOD CRC

entity toplevel is

```
port (
    operator          : in std_ulogic_vector(1 downto 0);
    clk,rst,load      : in std_ulogic;
    Din               : in std_ulogic_vector(3 downto 0);
    Datain            : in std_ulogic_vector(6 downto 0);
    syndrome,shift0,shift1,shift2,reg : out std_ulogic;
    result            : out std_ulogic_vector(6 downto 0)
);
```

end toplevel;

architecture toplevel_arch of toplevel is

```
component crc_encod    --encoder untk crc
port ( Clk,Rst,load    : in std_ulogic;
        Din            : in std_ulogic_vector(3 downto 0);
        CRC_Sum        : out std_ulogic_vector(6 downto 0) );
end component;
```


component crc_decode

```
port (
    Clk,Rst,load : in std_ulogic;

    Datain : in std_ulogic_vector(6 downto 0);

    syndrome : out std_ulogic;

    shift0 : out std_ulogic;
    shift1 : out std_ulogic;
    shift2 : out std_ulogic;
    reg : out std_ulogic;
    XCRC : out std_ulogic);
```

end component;

component mux

```
port (
    operator : in std_ulogic_vector(1 downto 0);
    CRC_en : in std_ulogic_vector(6 downto 0);
    CRC_de : in std_ulogic;

    result : out std_ulogic_vector(6 downto 0)
);
```

end component;

signal s_CRC_SUM : std_ulogic_vector(6 downto 0);

signal s_OutCRC : std_ulogic;

begin

U1: crc_encod port map (Clk=> clk,

Rst=>Rst,

load => load,

Din => din,

CRC_SUM=>s_CRC_SUM);

U2: crc_decode port map (

Clk => clk,

Rst => Rst,

load => load,

Datain => Datain,

syndrome =>syndrome,

shift0=>shift0,

shift1 =>shift1,

shift2=>shift2,

reg =>reg,

XCRC=> s_OutCRC);

U3: mux port map

(

operator=> operator,

CRC_en =>s_CRC_SUM,

CRC_de =>s_OutCRC,

result=> result);

Begitu juga dengan modul penggabungan yang kedua ini di mana pin-pin inputnya terdiri daripada masukan input untuk submodul pengekod CRC, input masukan untuk submodul peyahkod CRC, dan operator. Sementara outputnya terdiri daripada hasil keluaran submodul pengekod dan peyahkod . Apabila nilai input

dimasukkan di dalam “testbench”nya, ia akan melalui nilai dalam salah satu komponennya dan seterusnya dipetakan ke dalam senibina “top level”.

BAB ENAM

PENGUJIAN SISTEM

Dalam hal-hal ini proses pengujian sistem di lakukan ke atas sebuah model Hamming dan CRC. "Testbench" adalah satu bagian kod pengaturcaraan yang khusus yang digunakan dengan memasukkan nilai-nilai data input ke dalam komponen yang ingin diuji. Nilai data input yang akan dimasukkan ke dalam "testbench" akan di berikan ke dalam komponen yang di test. Hasil output untuk setiap komponen atau model yang diuji akan di simpan dalam sebuah variabel. Nilai output ini akan dibandingkan dengan nilai teori untuk memastikakan pengaturcaraan yang dihasilkan memberi nilai-nilai yang betul.

BAB ENAM

6.1 PENGUJIAN MODEL HANGSUNG

architecture stimulus of testbench is

component hamming;

port

datain : in std_logic_vector(3 downto 0);

hamming out std_logic_vector(3 downto 0);

PENGUJIAN SISTEM

end component;

signal datain : std_logic_vector(3 downto 0);

signal hamout : std_logic_vector(3 downto 0);

begin

HUT: hamming port map(datain,hamout);

stimulus1: process

6.0 PENGUJIAN SISTEM

Dalam bahagian ini proses pengujian sistem dilakukan ke atas kedua-dua modul Hamming dan CRC. “Testbench” adalah satu bahagian kod pengaturcaraan yang berbeza yang digunakan dengan memasukkan nilai-nilai data input ke dalam komponen yang ingin diuji. Nilai data input yang akan dimasukkan ke dalam “testbench” akan dipetakan ke dalam komponen yang dikehendaki. Output untuk setiap komponen atau modul yang diuji akan dikeluarkan melalui proses simulasi. Nilai output ini akan dibandingkan dengan nilai teori untuk melihat adakah pengaturcaraan yang dihasilkan menepati nilai-nilai teori tersebut.

6.1 PENGUJIAN SUBMODUL HAMMING

architecture stimulus of testbench is

component hamenc

port(

 datain : in std_ulogic_vector(3 downto 0);

 hamout : out std_ulogic_vector(6 downto 0)

);

end component;

signal datain : std_ulogic_vector(3 downto 0);

signal hamout : std_ulogic_vector(6 downto 0);

begin

DUT: hamenc port map(datain,hamout);

stimulus1: process

begin

DUT: hamenc port map(datain,hamout);

stimulus1: process

begin

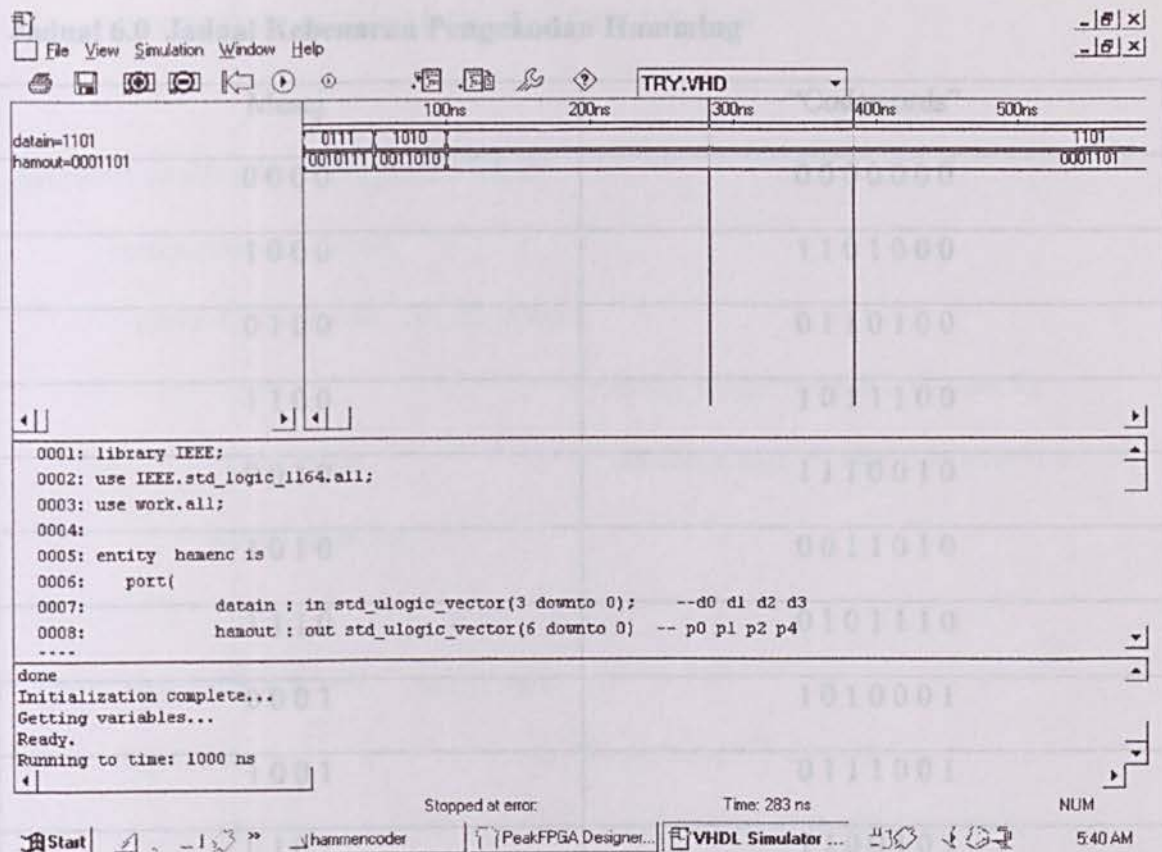
datain <= "0111";

wait for 50 ns;

wait;

end process;

Daripada pengekodean di atas, nilai datain adalah nilai input untuk 4 bit mesej yang dimasukkan secara selari. Kod pengaturcaraan untuk “testbench” ini akan dihubungkan bersama kod pengaturcaraan untuk submodul pengekod Hamming. Nilai input tersebut akan dipetakan melalui langkah pengaturcaraan pada submodul pengekod Hamming dan seterusnya akan menghasilkan 7 bit data output.



Rajah 6.0 Simulasi submodul pengekod Hamming

Daripada rajah 6.0 di atas adalah rajah dari proses simulasi untuk pengekodan Hamming. Dua data masukan 0111, dan 1010 yang diuji diwakilkan dengan nilai `datain`. Apabila ia disimulasi, hasil keluaran outputnya yang diwakili oleh pembolehubah `hamout`, adalah 0010111 dan 0011010. hasil ini akan dibandingkan dengan nilai teori seperti jadual di sebelah.

Berdasarkan nilai input yang dimasukkan, nilai data input dan keluaran yang dihasilkan adalah sama dengan nilai yang terdapat dalam jadual kebenaran 6.0. Maka submodul pengekod ini berfungsi seperti yang dikehendaki.

Jadual 6.0 Jadual Kebenaran Pengekoden Hamming

Mesej	"Codewords"
0 0 0 0	0 0 0 0 0 0 0
1 0 0 0	1 1 0 1 0 0 0
0 1 0 0	0 1 1 0 1 0 0
1 1 0 0	1 0 1 1 1 0 0
0 0 1 0	1 1 1 0 0 1 0
1 0 1 0	0 0 1 1 0 1 0
1 1 1 0	0 1 0 1 1 1 0
0 0 0 1	1 0 1 0 0 0 1
1 0 0 1	0 1 1 1 0 0 1
0 1 0 1	1 1 0 0 1 0 1
1 1 0 1	0 0 0 1 1 0 1
0 0 1 1	0 1 0 0 0 1 1
1 0 1 1	1 0 0 1 0 1 1
0 1 1 1	0 0 1 0 1 1 1
1 1 1 1	1 1 1 1 1 1 1

Berdasarkan nilai input yang dimasukkan, nilai data input dan keluaran yang dihasilkan adalah sama dengan nilai yang terdapat dalam jadual kebenaran 6.0. Maka submodul pengekod ini berfungsi seperti yang dikehendaki.

6.2 PENGUJIAN SUBMODUL PENGEKOD CRC

Architecture stimulus of crc_testbench is

```
component crc_encod
```

```
port ( Clk,Rst,load : in std_ulogic;
```

```
      Din                                     : in std_ulogic_vector(3 downto 0);
```

```
      CRC_Sum                                : out std_ulogic_vector(6 downto 0));
```

```
end component;
```

```
wait for 100 ns;
```

```
signal Clk,Rst,load : std_ulogic;
```

```
signal Din           : std_ulogic_vector(3 downto 0);
```

```
signal CRC_Sum       : std_ulogic_vector(6 downto 0);
```

```
Signal Clock_cycle   : natural := 0;
```

Begin

```
DUT: crc_encod Port Map (Clk,Rst,load,Din,CRC_Sum);
```

```
CLOCK: process
```

```
begin
```

```
  Clock_cycle <= Clock_cycle + 1;
```

```
  Clk <= '1';
```

```
  wait for 25 ns;
```

```
  Clk <= '0';
```

```
  wait for 25 ns;
```

```
end process;
```


Begin

```
Rst <= '1';
```

```
load <= '0';
```

```
    din <= "1011";
```

```
wait for 100 ns;
```

```
Rst <= '0';
```

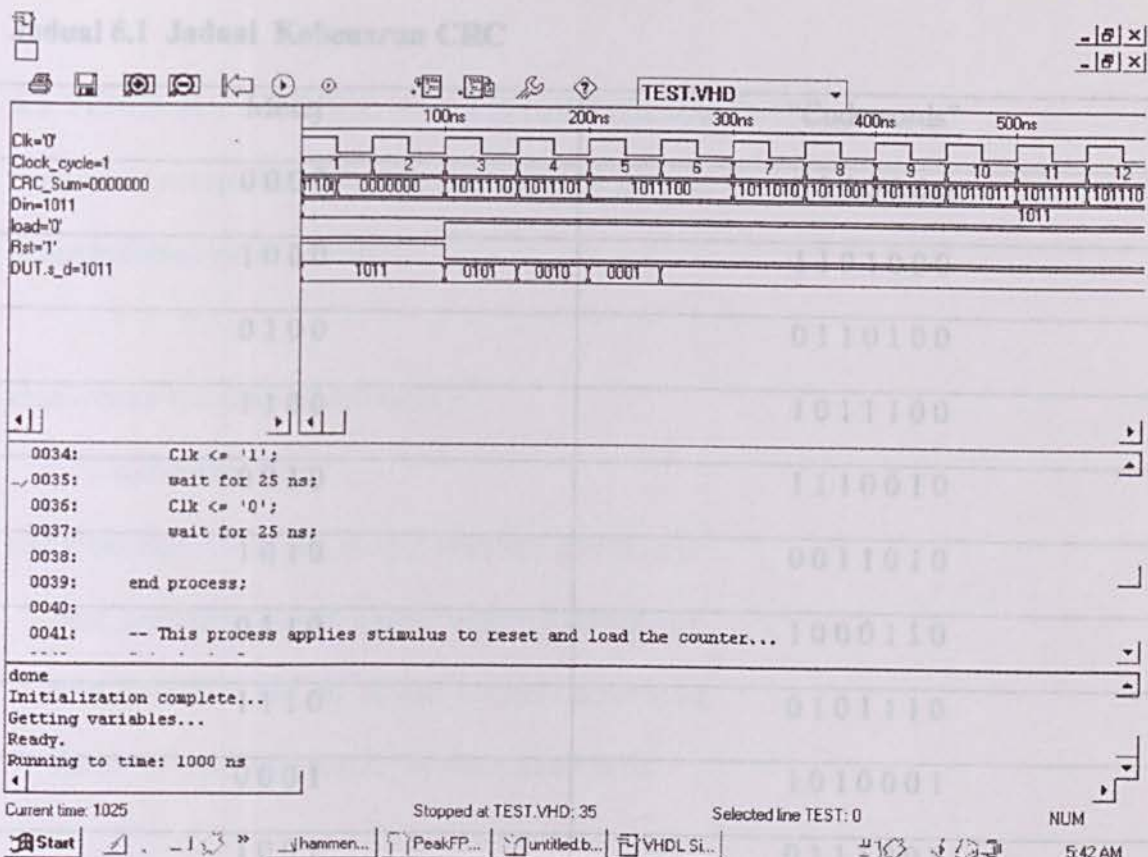
```
load <= '1';
```

```
    din <= "1011";
```

```
wait for 100 ns;
```

```
wait; end Process;
```

Seperti pengkodean Hamming di mana nilai data dimasukkan adalah 4 bit, dan ia akan menghasilkan keluaran data yang terenkod sebanyak 7 bit. Data hanya akan mula dimasukkan pada nilai load = '1' , dan nilai Rst='1'. Hasil simulasinya adalah seperti yang ditunjukkan pada mukasurat sebelah.



Rajah 6.1 Simulasi submodul pengekod CRC

Daripada rajah di atas, dapat dilihat bahawa nilai data input yang diwakili oleh Din adalah 1011. Dut.s.d adalah pemboleh ubah yang menunjukkan nilai anjakan setiap kali 1 bit daripada nilai Din dimasukkan ke dalam penimbal. CRCsum pula mewakili nilai data bit yang terenkod. Nilai terenkod di baca pada kitar jam yang ke enam iaitu 1011100. Nilai ini juga di bandingkan dengan nilai teori yang terdapat pada jadual kebenaran CRC seperti di sebelah.

Jadual 6.1 Jadual Kebenaran CRC

Mesej	“Codewords”
0 0 0 0	0 0 0 0 0 0 0
1 0 0 0	1 1 0 1 0 0 0
0 1 0 0	0 1 1 0 1 0 0
1 1 0 0	1 0 1 1 1 0 0
0 0 1 0	1 1 1 0 0 1 0
1 0 1 0	0 0 1 1 0 1 0
0 1 1 0	1 0 0 0 1 1 0
1 1 1 0	0 1 0 1 1 1 0
0 0 0 1	1 0 1 0 0 0 1
1 0 0 1	0 1 1 1 0 0 1
0 1 0 1	1 1 0 0 1 0 1
1 1 0 1	0 0 0 1 1 0 1
0 0 1 1	0 1 0 0 0 1 1
1 0 1 1	1 0 0 1 0 1 1
1 1 1 1	1 1 1 1 1 1 1

Berdasarkan nilai input yang dimasukkan, nilai data input yang dimasukkan iaitu 1011, menghasilkan nilai output 1011100 yang diwakili oleh pemboleh ubah CRCumnya. Jika disemak daripada jadual kebenaran CRC, nilai keluaran yang dihasilkan adalah sepadan sama dengan nilai mesejnya seperti yang terdapat dalam jadual kebenaran 6.1. Maka pengujian ke atas komponen ini telah berjaya.

6.3 PENGUJIAN MODUL MULTIPLEKSOR HAMMING

Pengujian multipleksor diuji dengan nilai data input seperti yang terdapat dalam kod pengaturcaraannya di bawah.

architecture stimulus of testbench is

```
component host
port (operator : in std_ulogic_vector(1 downto 0);
      host_hamout : in std_ulogic_vector(6 downto 0);
      host_DOUT : in std_ulogic_vector(3 downto 0);
      result : out std_ulogic_vector(6 downto 0)
    );
end component;

constant PERIOD: time := 50 ns;

signal operator : std_ulogic_vector(1 downto 0);
signal host_hamout : std_ulogic_vector(6 downto 0);
signal host_DOUT : std_ulogic_vector(3 downto 0);
signal result : std_ulogic_vector(6 downto 0);

begin

DUT: host port map(operator, host_hamout, host_DOUT, result);

INPUTS: process
```

```
begin
```

```
-- input encoder
```

```
operator<= "01";
```

```
host_hamout <= "1001011";
```

```
wait for PERIOD;
```

```
-- input decoder
```

```
-- operator<= "10";
```

```
-- host_DOUT<= "0111";
```

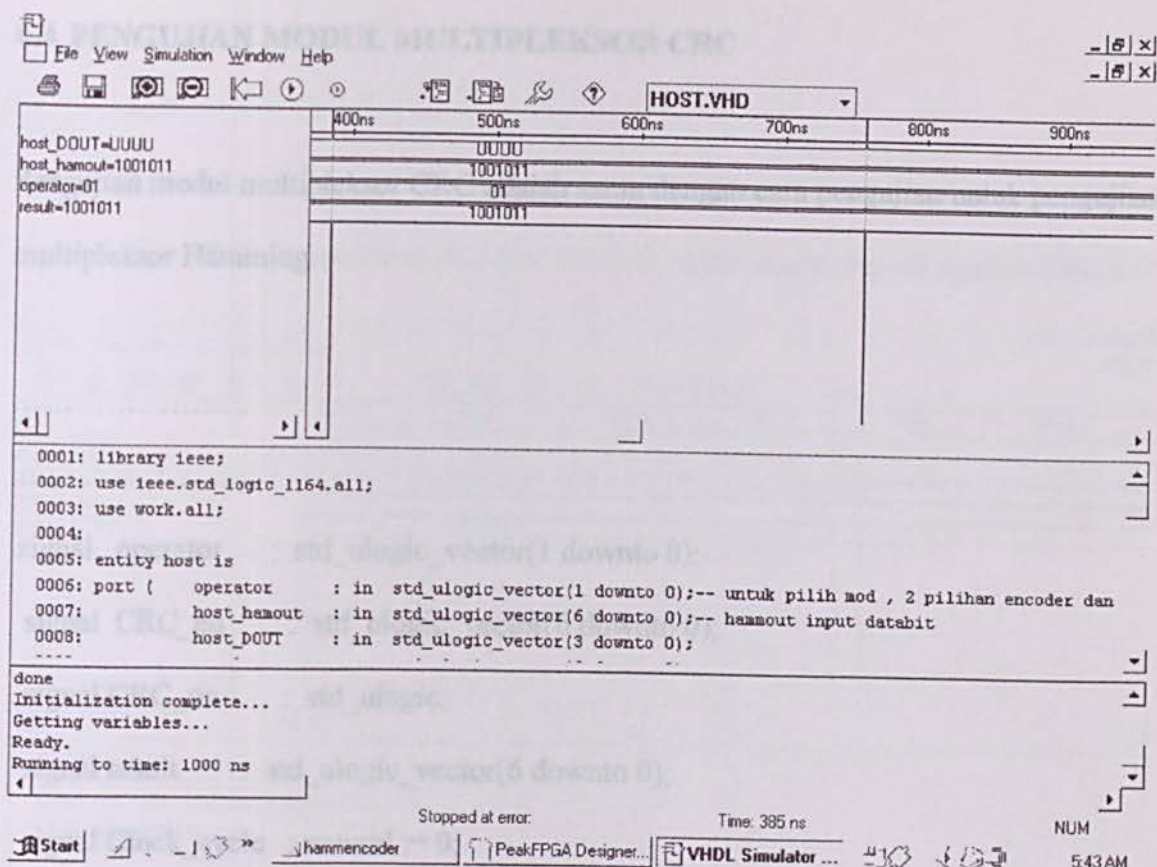
```
-- wait for PERIOD;
```

```
wait;
```

```
end process;
```

Rajah 4.2 Simulasi submodul multiplexsor Hamming

Dari kod pengaturcaraan ini, ia akan dihubungkan dengan kod pengaturcaraan submodul multipleksor , kemudian disimulasi.



Rajah 6.2 Simulasi submodul multipleksor Hamming

Rajah di atas menunjukkan proses simulasi untuk modul Hamming. Nilai operator yang dimasukkan adalah 01, hanya ouput untuk submodul pengekod Hamming dimasukkan iaitu 1001011. Pemboleh ubah “result” mewakili keluaran komponen multipleksor Hamming. Nilai tersebut adalah 1001011, iaitu nilai yang sama dengan masukan untuk nilai output Hamming iaitu 1001011, maka pengaturcaraan untuk modul ini berjaya.

6.4 PENGUJIAN MODUL MULTIPLEKSOR CRC

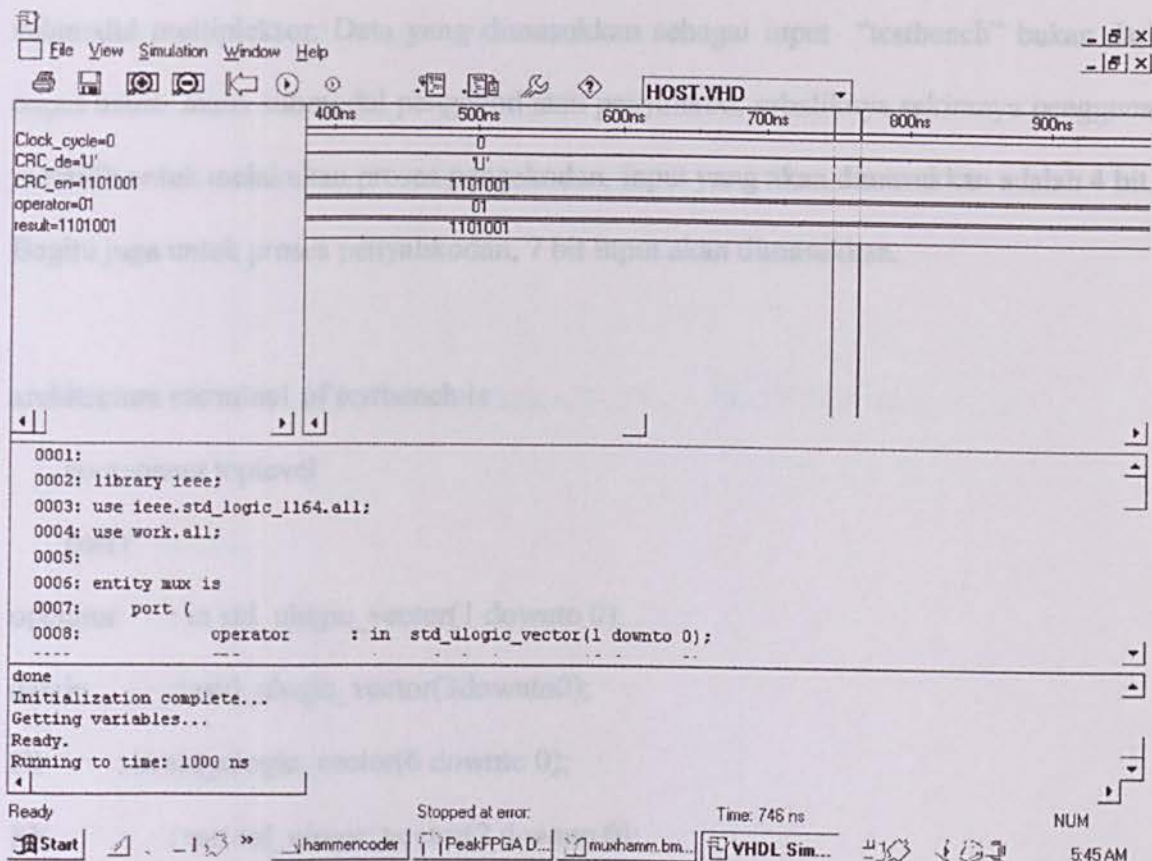
Pengujian modul multipleksor CRC adalah sama dengan cara pengujian untuk pengujian multipleksor Hamming.

```
.....  
.....  
signal operator    : std_ulogic_vector(1 downto 0);  
signal CRC_en      : std_ulogic_vector(6 downto 0);  
signal CRC_de      : std_ulogic;  
signal result      : std_ulogic_vector(6 downto 0);  
signal Clock_cycle : natural := 0;  
  
begin  
    DUT: mux port map (operator,CRC_en,CRC_de,result);  
    stimulus1: process  
    begin  
        -- input encoder  
        operator <= "01";  
        CRC_en <= "1101001";  
        wait for 25 ns;  
        -- input decoder  
        --operator <= "10";
```

```
--CRC_de <= '1';
```

```
wait;
```

Dari kod pengaturcaraan ini, ia akan dihubungkan dengan kod pengaturcaraan modul multiplexsor CRC , kemudian disimulasi. Hasil simulasi adalah seperti rajah di bawah.



Rajah 6.3 : Simulasi submodul multiplexsor CRC

Daripada rajah 6.3 di atas, nilai operator yang dipilih adalah 01, maka output daripada submodul pengekodan CRC akan dimasukkan sebagai input kepada modul multiplexsor. Nilai CRC_en menunjukkan nilai data input yang dimasukkan iaitu 1101001. Pembolehubah result mewakili nilai keluaran untuk modul multiplexsor. Nilai keluarannya adalah sama dengan output pengekodan CRC. Maka pengaturcaraan untuk modul ini telah berjaya.

6.5 PENGUJIAN MODUL HAMMING.

Pengujian modul Hamming bertujuan untuk menguji kesepaduan antara semua sub-modul yaitu submodul pengekod Hamming, submodul penyahkod Hamming dan submodul multipleksor. Data yang dimasukkan sebagai input “testbench” bukan dari output mana- mana submodul pengekod atau penyahkod, sebaliknya sekiranya pengguna memilih untuk melakukan proses pengkodean, input yang akan dimasukkan adalah 4 bit. Begitu juga untuk proses penyahkodan, 7 bit input akan dimasukkan.

architecture stimulus1 of testbench is

```
    component toplevel
    port (
operator      : in std_ulogic_vector(1 downto 0);
datain        : in std_ulogic_vector(3 downto 0);
DI            : in std_ulogic_vector(6 downto 0);
SY            : out std_ulogic_vector(2 downto 0);
DO            : out std_ulogic_vector(6 downto 0)
DOU           : out std_ulogic_vector(6 downto 0);
result        : out std_ulogic_vector(6 downto 0)
    );

    end component;
```

```
constant PERIOD : time := 50 ns;
```



```

signal operator : std_ulogic_vector(1 downto 0);

signal datain   : std_ulogic_vector(3 downto 0);          signal DI
               : std_ulogic_vector(6 downto 0);    signal SY      : std_ulogic_vector(2
downto 0);

signal DO       : std_ulogic_vector(6 downto 0);
signal DOU      : std_ulogic_vector(6 downto 0);
signal result   : std_ulogic_vector(6 downto 0);

begin

DUT: toplevel port map (operator,datain,DI,SY, DO,DOU,result);

input : process
begin
    -- input encoder
    operator<= "01";
    datain<= "1101";
    wait for PERIOD;

    -- input decoder
    -- operator<= "10"; -- tiada ralat
    -- DI<= "0111010";
    -- wait for PERIOD;

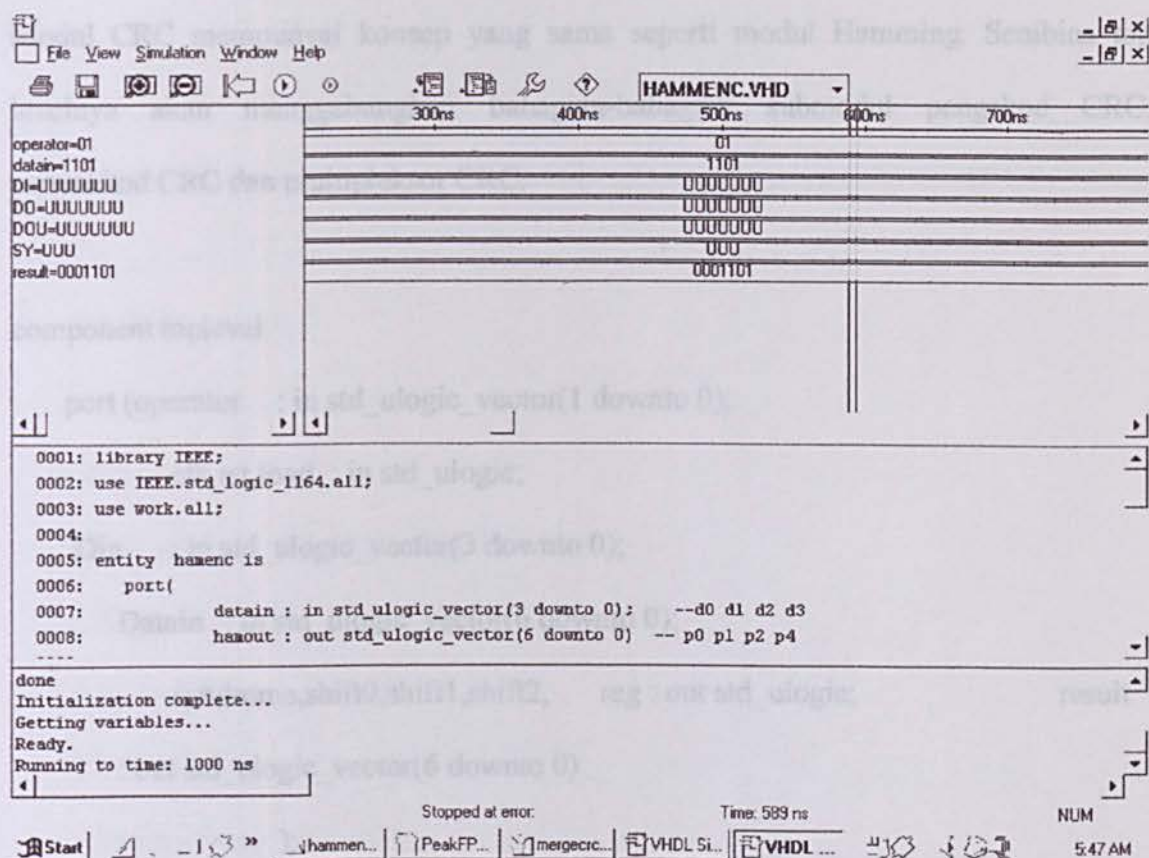
    -- operator<= "10"; -- ralat

```

```
-- DI<="1101101"; --> menjadi 1001011; Output daripada submodul pengkod
-- wait for PERIOD; --> akan melalui pengaliran ke submodul multiplexer maka
wait; --> akan menghasilkan nilai 1001011 sebagai output untuk modul Hamming
end process;
```

Selanjutnya akan membuat bit akan dipetakan pada komponen-komponen "top level" yang mewakili nilai input dan outputnya.

Dari kod pengaturcaraan ini, ia akan dihubungkan dengan kod pengaturcaraan untuk modul Hamming, kemudian disimulasi.



Rajah 6.3: Simulasi modul Hamming

Rajah di sebelah menunjukkan proses simulasi untuk modul Hamming. Input yang dimasukkan bernilai 1011. Nilai input ini akan melalui kod pengaturcaraan pengkodan

Hamming dan akan dikodkan menjadi 1001011. Output daripada submodul pengekod Hamming bernilai 1001011 akan melalui pengaturcaraan submodul multipleksor maka operator akan mengeluarkan nilai 1001011 sebagai output untuk modul Hamming. Semua nilai keluaran dan masukan bit akan dipetakan pada komponen- komponen “top level” yang mewakili nilai input dan outputnya.

6.6 PENGUJIAN MODUL CRC

Modul CRC mempunyai konsep yang sama seperti modul Hamming. Senibina top levelnya akan menggabungkan bahagian-bahagian submodul pengekod CRC, penyahkod CRC dan multipleksor CRC.

component toplevel

```
port (operator : in std_ulogic_vector(1 downto 0);
```

```
clk,rst,load : in std_ulogic;
```

```
Din : in std_ulogic_vector(3 downto 0);
```

```
Datain : in std_ulogic_vector(6 downto 0);
```

```
syndrome,shift0,shift1,shift2, reg : out std_ulogic; result
```

```
: out std_ulogic_vector(6 downto 0)
```

```
);
```

```
end component;
```

```
constant PERIOD : time := 50 ns;
```

```
signal operator : std_ulogic_vector(1 downto 0);
```



```

signal clk,rst,load : std_ulogic;

signal Din          : std_ulogic_vector(3 downto 0);

signal Datain       : std_ulogic_vector(6 downto 0);

signal syndrome,shift0,shift1,shift2,reg : std_ulogic;

signal result       : std_ulogic_vector(6 downto 0);

signal Clock_cycle   : natural := 0;

begin

DUT:toplevelportmap(operator,clk,rst,load,Din,Datain,syndrome,shift0,shift1,shift2,reg,
result);

    CLOCK: process

begin
    Clock_cycle <= Clock_cycle + 1;

    Clk <= '1';

    wait for 25 ns;

    Clk <= '0';

    wait for 25 ns;

end process;

INPUTS: process

begin

```

```
--input CRC encoder
```

```
Rst <= '1';
```

```
load <= '0';
```

```
operator <= "01";
```

```
Din <= "1101";
```

```
wait for 100 ns;
```

```
Rst <= '0';
```

```
load <= '1';
```

```
operator <= "01";
```

```
Din <= "1101";
```

```
wait for 100 ns;
```

```
-- input CRC decoder
```

```
rst<='1';
```

```
load<='0';
```

```
operator <= "10";
```

```
Datain <= "0110001";
```

```
wait for PERIOD;
```

```
rst<='0';
```

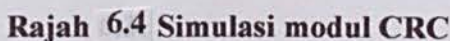
```
load<='1';
```

```
operator <= "10";
```

```
Datain <= "0110001";
```

```
wait for PERIOD;
```

Dari kod pengaturcaraan ini, ia akan dihubungkan dengan kod pengaturcaraan untuk modul CRC, kemudian disimulasi.



71

melalui komponen pengekod CRC. 7 bit output yang dihasilkan iaitu 0001101, akan dihantar ke operator. Nilai operator akan disetkan kepada 01 untuk nilai pengekod dan output akan dikeluarkan.

BAB TUJUH

PERBINCANGAN

7.0 PERBINCANGAN

Bab ini akan membincangkan tentang kelemahan dan kelebihan sistem, kekurangan yang di hadapi sistem, membolehkan projek ini. Peryelesaian ke atas setiap masalah dan perubahan yang telah dilakukan juga akan ditulis dalam bab ini.

7.1 PERUBAHAN BERKESKUTAN AWAL SISTEM

Pada awalnya sistem ini terdiri daripada dua modul utama, iaitu modul pengkod dan modul pemroses. Modul pengkod terdiri daripada operator, pemroses, pengkod CRC, dan modul pemroses terdiri daripada pemroses Hamming dan pemroses CRC. Tetapi pada masa ini, sistem ini telah diubah menjadi satu modul utama iaitu modul Hamming yang terdiri daripada pemroses Hamming. Dan modul yang lain iaitu modul CRC yang terdiri daripada operator, pengkod dan pemroses CRC. Reka bentuk yang terbaru adalah seperti yang ditunjukkan di bawah.

PERBINCANGAN

7.0 PERBINCANGAN

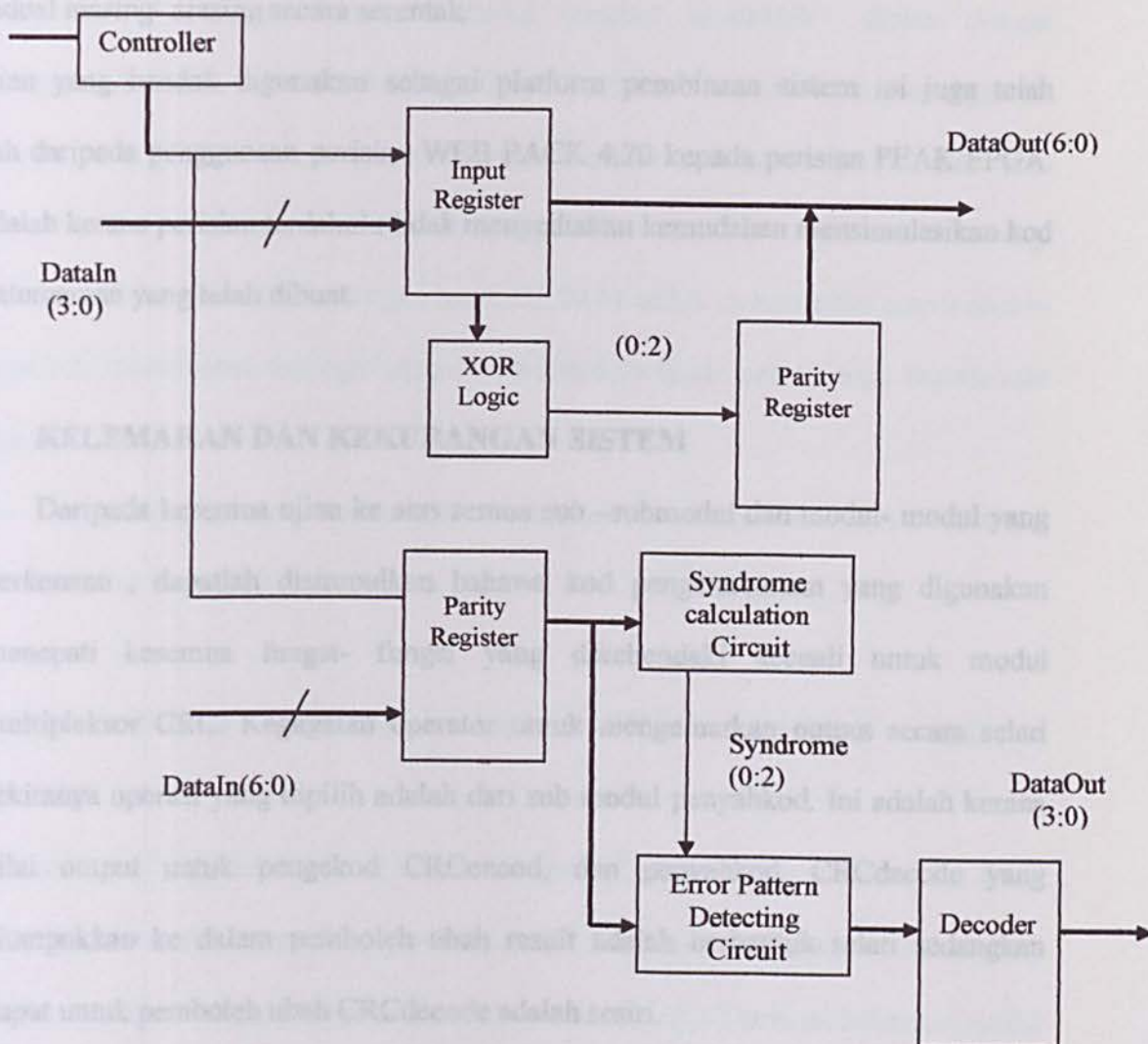
Bab ini akan membincangkan tentang kelemahan dan kelebihan sistem, kekangan yang di hadapai semasa menjalankan projek ilmiah ini. Penyelesaian ke atas setiap masalah dan perubahan yang telah dilakukan juga akan ditulis dalam bab ini.

7.1 PERUBAHAN REKABENTUK AWAL SISTEM

Pada awalnya, sistem ini terdiri daripada dua modul utama, iaitu modul pengekod dan modul penyahkod. Modul pengekod terdiri daripada operator, submodul pengekod Hamming dan submodul pengekod CRC, dan modul penyahkod terdiri daripada penyahkod Hamming dan penyahkod CRC. Tetapi pada saat akhir sistem tersebut telah di ubah kepada dua modul utama iaitu modul Hamming yang terdiri daripada operator, pengekod dan penyahkod Hamming. Dan modul yang kedua iaitu modul CRC yang terdiri daripada operator, pengekod dan penyahkod CRC. Rekabentuk yang terbaru adalah seperti yang ditunjukkan di sebelah.

Rajah 7.0 : Litar awal diagram aras tertinggi untuk keseluruhan sistem

Selain daripada itu, pada peringkat awal modul pengekod Hamming menggunakan penyahkod binary, kemudian komponen tersebut telah di ubah kepada penyahkod binary kerana proses yang dilalui oleh 4 bit input untuk menghasilkan 7 bit pengkodan adalah



Rajah 7.0 : Lakaran semula diagram aras tertinggi untuk keseluruhan sistem

Selain daripada itu, pada peringkat awal modul pengkod Hamming menggunakan penjadual anjakan, kemudian komponen tersebut telah di ubah kepada penjadual biasa kerana proses yang dilalui oleh 4 bit input untuk menghasilkan 7 bit pengkodan adalah proses pengkodan adalah 72 bit.

segerak. Ini bermakna semua penempatan 4 bit input maklumat diletakkan ke dalam penjadual masing- masing secara serentak.

Perisian yang hendak digunakan sebagai platform pembinaan sistem ini juga telah diubah daripada penggunaan perisian WEB PACK 4.20 kepada perisian PEAK FPGA. Ini adalah kerana perisian terdahulu tidak menyediakan kemudahan mensimulasikan kod pengaturcaraan yang telah dibuat.

7.2 KELEMAHAN DAN KEKURANGAN SISTEM

Daripada kesemua ujian ke atas semua sub –submodul dan modul- modul yang berkenaan , dapatlah disimpulkan bahawa kod pengaturcaraan yang digunakan menepati kesemua fungsi- fungsi yang dikehendaki kecuali untuk modul multiplexsor CRC. Kegagalan operator untuk mengeluarkan output secara selari sekiranya operasi yang dipilih adalah dari sub modul penyahkod. Ini adalah kerana nilai output untuk pengekod CRCencod, dan penyahkod, CRCdecode yang diumpukkan ke dalam pemboleh ubah result adalah berbentuk selari sedangkan ouput untuk pemboleh ubah CRCdecode adalah sesiri.

Sebagai penyelesaiannya, perlu disediakan penimbal untuk nilai selari (6 downto 0), dan menggunakan kaedah anjakan untuk memuatkan kesemua output nilai bit sesiri ternyahkod daripada sub modul penyahkod CRC. –

Adalah menjadi kebiasaan untuk setiap sistem mempunyai kelemahan dan kelebihan tersendiri. Kelemahan sistem ini ialah jumlah bit yang dimasukkan adalah terlalu kecil, sedangkan pada waktu sekarang jumlah bit yang dimasukkan untuk proses pengekodan adalah 32 bit .

Yang kedua ialah modul CRC dan modul Hamming tidak perlu dibina secara berasingan. Sepatutnya kesemua modul tersebut hendaklah dibina dengan menggunakan satu rekabentuk sahaja. Cuma perlukan 1 operator yang mempunyai empat mod pilihan untuk memilih samada untuk melakukan pengekodan atau penyahkodan CRC atau melakukan pengekodan atau penyahkodan Hamming. Operator boleh diwakilkan dengan bit 00,01,10,11 untuk menandakan empat pilihan tersebut. Jumlah operator juga dapat dikurangkan daripada tiga operator kepada satu operator sahaja.

Penggabungan antara kedua- dua modul Hamming dan CRC juga tidak dapat dilaksanakan kerana kekangan masa.

7.3 PERBANDINGAN MODUL HAMMING DENGAN SISTEM SEBELUMNYA

Perbandingan juga telah dibuat antara sistem ini dengan sistem “Hamming Code Error Detector/Decoder” yang dibangunkan Jeniffer Delliner and Melissa De Priest yang diperolehi dari Internet akan dilaksanakan dalam cip FPGA. Terdapat beberapa modul yang dibangunkan dalam sistem ini iaitu litar pendarab matriks, litar penyahkod, modul H- matriks dan sebagainya sebagaimana yang ditunjukkan dalam 7.0 di sebelah. Perbandingan di buat antara modul- modul yang terdapat dalam sistem kami dengan modul –modul sistem yang dibandingkan seperti yang terdapat dalam jadual disebelah.

Jadual 7.0: Perbandingan antara sistem pengesanan dan pembetulan ralat (B) dengan sistem pegesan dan penyahkod ralat (A).

Modul A	Modul B	Fungsi
Litar pendarab matriks	Operasi get XOR	Menjana fungsi sindrom
Litar pengesan ralat	Operasi get XOR	Mengesan sekiranya ralat wujud pada mana-mana bahagian pada 7 bit yang dihantar.
Litar penyahkod	Penimbal	Mengasingkan 3 bit pariti dengan 4 bit maklumat.

Merujuk kepada jadual di atas, sebenarnya perbandingan di buat antara komponen yang terdapat dalam submodul penyahkod Hamming. Dalam modul A terdapat beberapa litar yang digunakan untuk fungsi penyahkodan, . Diantaranya ialah litar pendarab matriks, litar pengesan ralat dan litar penyahkod. Fungsi sistem tersebut adalah sama dengan modul penyahkod iaiautu ia akan menerima 7 input untuk dinyahkodkan kepada 4 bit output. Diantara perbezaan yang di dapati ialah:-

1. Pengiraan sindrom

Di dalam modul penyahkod yang di bangunkan dalam bahagian berasingan daripada yang dibangunkan di dalam sistem ini, terdapat pengiraan nilai – nilai sindrom yang dijana daripada 7 bit input yang diterima. 3 bit sindrom ini adalah bertujuan untuk menentukan samada terdapat ralat aatau tidak di dalam 7 bit input yang diterima. Kesemua ketiga – tiga bit sindrom yang dijana mestilah 0. Bit- bit sindrom ini akan dikeluarkan satu persatu untuk didarabkan dengan 7 bit input yang dimasukkan. Sekiranya terdapat salah satu daripada bit- bit sindrom ini bernilai 1, maka pasti terdapat ralat dalam 7 bit input yang diterima. Sistem A menggunakan aplikasi pendaraban H matriks yang digunakan dalam litar pendarab matriks untuk menghasilkan 3 bit sindrom. H matiks adalah terdiri daripada 3 X 7 nilai bit. Nilai untuk H matriks ini boleh adalah dijana daripada penggabungan matriks identiti dan nilai – nilai bit yang bukan pariti yang mewakili kedudukan 3, 5, 6, dan 7 dalam input yang dimasukkan. 21 penimbal digunakan untuk memuatkan kesemua bit- bit tersebut. Pengawal pula diperlukan untuk memastikan setiap bit sindrom di darabkan dengan 7 bit dari baris yang pertama nilai- nilai bit yang terdapat dalam matriks H. Berbanding dengan submodul pengekod dalam sistem kami yang menggunakan perlaksanaan get XOR untuk menghasilkan 3 bit sindrom tersebut secara selari. Proses ini tidak memerlukan pengawal untuk mengawal aliran bit seperti yang terdapat dalam sistem A.

2. Litar penyahkod

Setelah bit sindrom dihasilkan, litar penyahkod bertujuan untuk mengesan kedudukan ralat yang terdapat dalam 7 bit input tersebut. Jika kedudukan ralat dapat di kesan, maka isyarat akan dihantar untuk meminta sumber dihantar semula. Tiada fungsi pembetulan ralat yang dilakukan dalam sistem A. Berlainan dengan sistem kami yang

hanya menggunakan pelaksanaan get XOR untuk fungsi penyahkodan. Ia akan membetulkan ralat pada kedudukan ralat yang dikesan.

3. Kompleksiti

Terlalu banyak subsistem yang kompleks yang terdapat dalam modul A untuk fungsi penyahkodan berbanding sistem yang ingin dibangunkan. Berbanding dengan sistem kami yang hanya menggunakan pelaksanaan get XOR untuk tujuan yang sama tetapi berupaya untuk membetulkan ralat tersebut. Sistem A hanya mampu mengesan kedudukan ralaat, dan sekiranya ralat wujud, ia akan menghantar isyarat untuk menerima input yang sama untuk kali kedua tetapi tidak membetulkannya.

Sebagai kesimpulannya, projek ini telah berjaya disiapkan kerana penggabungan modul Hamming dan CRC yang hendak diletakkan dalam satu rekabentuk yang sama adalah fasa tambahan. Penggunaan kedua-dua teknik Hamming dan CRC yang terdapat dalam sistem ini sebenarnya adalah salingan, kerana Hamming walaupun mampu membetulkan ralat tetapi tidak dapat mengesan lebih daripada dua ralat pada satu amsa yang sama. Berbanding dengan teknik CRC walaupun tidak berupaya membetulkan ralat sebaliknya mampu untuk mengesan bit ralat yang lebih besar nilainya daripada Hamming secara serentak.

MANUAL PENGGUNA

Manual ini akan menunjukkan tentang bagaimana pin-pin, dan komponen-komponen yang terlihat dalam produksi sistem ini.

1.0 Diagram Area Tertinggi Keterangan Sistem



APPENDIKS

Table 1.0 Rincian Diagram Sistem Pengawasan dan Perencanaan Ralat

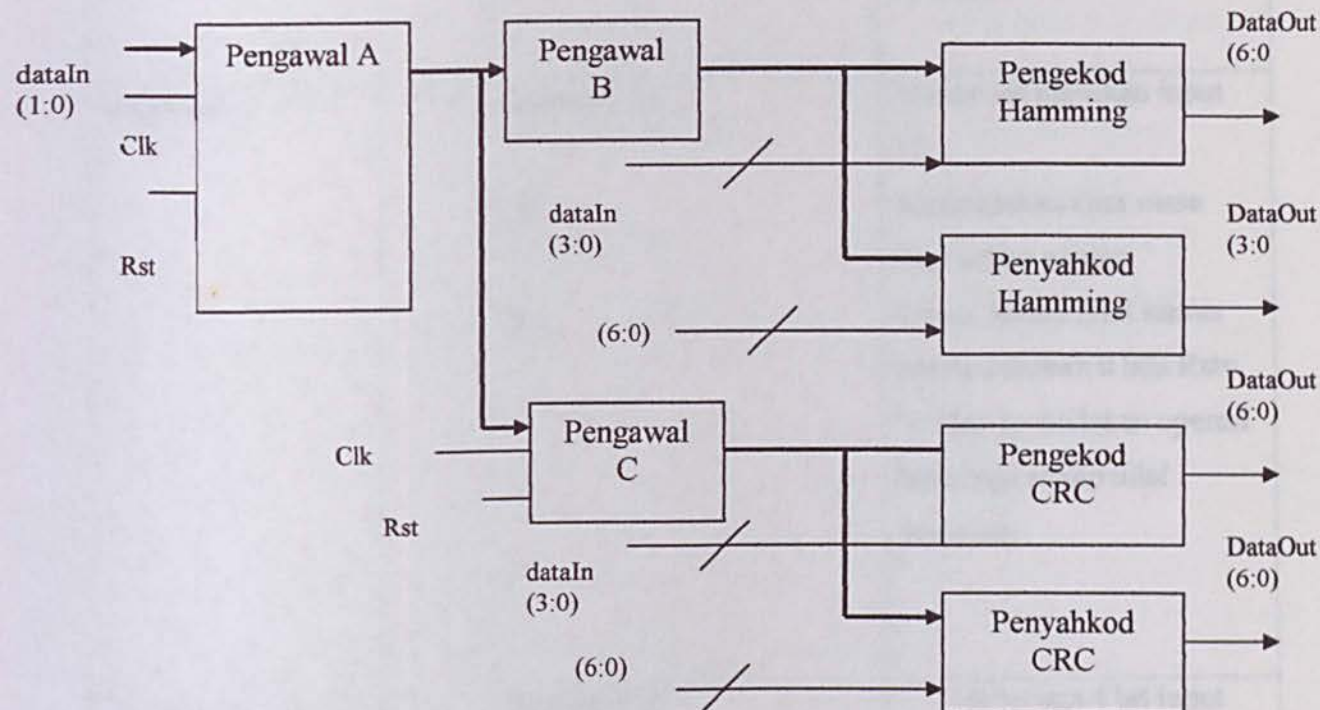
Index 1.0 Fungsi Keterangan komponen dalam sistem

Subsistem	Peta	Fungsi
Pengawal	Ra	Untuk arahan sistem secara pambatalan 6 bit Ral- "P" dan membolehkan operasi baru bagi setiap nilai

MANUAL PENGGUNA

Manual ini akan menerangkan tentang kesemua pin- pin, dan komponen- komponen yang terlibat dalam pembinaan sistem ini.

1.0 Diagram Aras Tertinggi Keseluruhan Sistem



Rajah 1.0 Blok Diagram Sistem Pengesanan dan Pembetulan Ralat

Jadual 1.0 Fungsi keseluruhan komponen dalam sistem

Submodul	Pin	Fungsi
Pengawal	Rst	Untuk setkan nilai semua pombolehubah 0 bila Rst= '1' dan memulakan operasi baru bagi setiap nilai

	DataOut(3:0) Clk	masukan . Menandakan kitar masa bagi setiap operasi
Pengekod CRC	DataIn (3:0) DataIn (1:0) DataIn (1:0) DataOut(6:0)	Pilihan mod samada 01 untuk modul Hamming dan 10 untuk modul CRC Mengeluarkan 7 bit
Pengawal B	DataIn (1:0)	Menerima nilai masukan operator 01
Penyahkod CRC	DataIn (6:0)	Menerima 6 bit input
Pengawal C	DataIn (1:0) DataIn (1:0) Clk Rst	Menerima masukan input 10 Menandakan kitar masa bagi setiap operasi Untuk setkan nilai semua pombolehubah 0 bila Rst= '1' dan memulakan operasi baru bagi setiap nilai masukan .
Pengekod Hamming	DataIn (3:0) DataIn (1:0) DataOut(6:0)	Menerima 4 bit input masuk. Menerima isyarat operator ="01" Mengeluarkan 7 bit output terenkod
Penyahkod Hamming	DataIn (6:0) DataIn (1:0)	Menerima 6 bit input masuk Menerima isyarat operator="10"

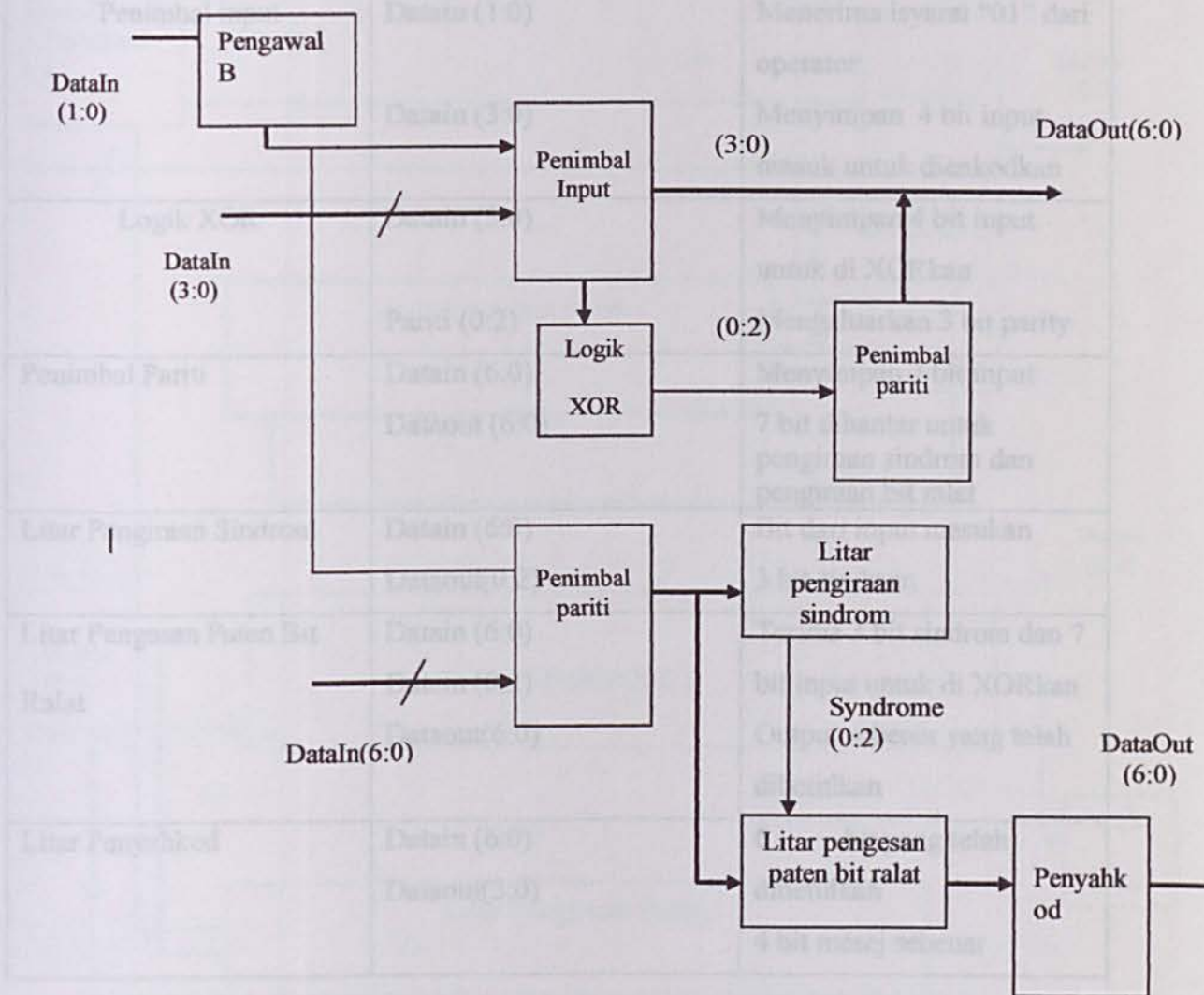
	Dataout(3:0)	Mengeluarkan 4 bit output ternyahkod
Pengekod CRC	Datain (3:0) Datain (1:0) DataOut(6:0)	Menerima 4 bit input masuk. Menerima isyarat operator ="01" Mengeluarkan 7 bit output terenkod
Penyahkod CRC	Datain (6:0) Datain (1:0) Dataout(3:0)	Menerima 6 bit input data masuk Menerima isyarat operator ="10" Mengeluarkan 4 bit output ternyahkod

Rajah 2.6 Blok Diagram Modul Hamming

Jadual 2.1 Fungsi Keseluruhan Komponen Dalam Modul Hamming

Submodul	Pin	Fungsi
Register 6	Datain (1:0)	Pilihan mod operasi 0 atau

2.0 Blok Diagram Modul Hamming



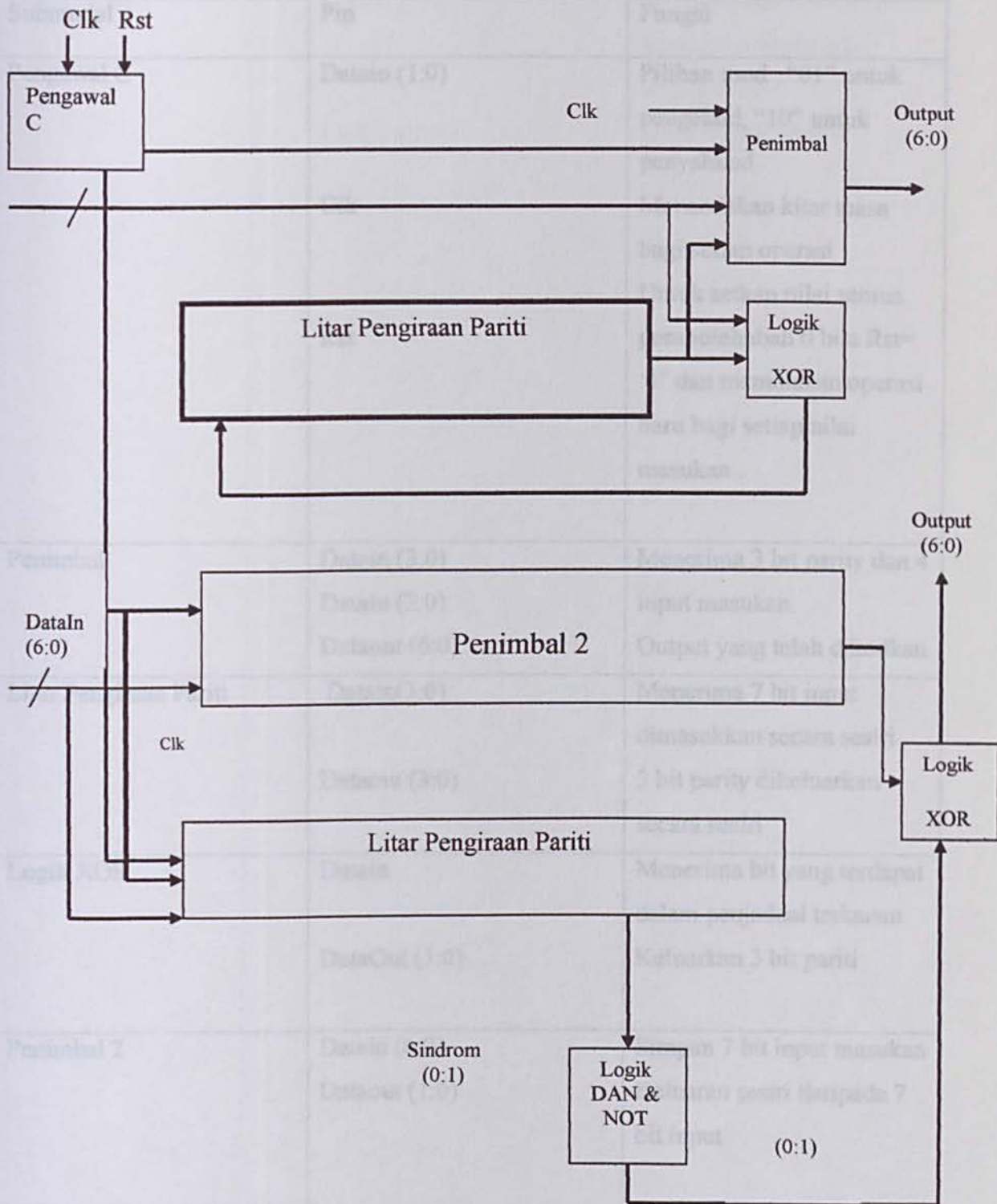
Rajah 2.0 Blok Diagram Modul Hamming

Jadual 2.0 : Fungsi Keseluruhan Komponen Dalam Modul Hamming

Submodul	Pin	Fungsi
Pengawal B	Datain (1:0)	Pilihan mod samada 01 atau

		10 untuk modul pengkode atau penyahkod Hamming
Penimbal input	Datain (1:0) Datain (3:0)	Menerima isyarat "01" dari operator Menyimpan 4 bit input masuk untuk dienkodkan
Logik XOR	Datain (3:0) Pariti (0:2)	Menyimpan 4 bit input untuk di XORkan Mengeluarkan 3 bit parity
Penimbal Pariti	Datain (6:0) Dataout (6:0)	Menyimpan 7 bit input 7 bit dihantar untuk pengiraan sindrom dan pengiraan bit ralat
Litar Pengiraan Sindrom	Datain (6:0) Dataout(0:2)	Bit dari input masukan 3 bit sindrom
Litar Pengesan Paten Bit Ralat	Datain (6:0) Datain (0:2) Dataout(6:0)	Terima 3 bit sindrom dan 7 bit input untuk di XORkan Output sebenar yang telah dibetulkan
Litar Penyahkod	Datain (6:0) Dataout(3:0)	6 bit yang telah dibetulkan 4 bit mesej sebenar

3.0 Blok Diagram Modul CRC



Rajah 3.0 Diagram Aras Tertinggi Untuk Modul CRC

Jadual 3.0: Fungsi keseluruhan Komponen Dalam Modul CRC

Submodul	Pin	Fungsi
Pengawal C	Datain (1:0) Clk Rst	Pilihan mod , "01" untuk pengekod, "10" untuk penyahkod Menandakan kitar masa bagi setiap operasi Untuk setkan nilai semua pombolehubah 0 bila Rst= '1' dan memulakan operasi baru bagi setiap nilai masukan .
Penimbal	Datain (3:0) Datain (2:0) Dataout (6:0)	Menerima 3 bit parity dan 4 input masukan Output yang telah dikodkan
Litar Pengiraan Pariti	Datain(3:0) Dataout (3:0)	Menerima 7 bit input dimasukkan secara sesiri 3 bit parity dikeluarkan secara sesiri
Logik XOR	Datain DataOut (3:0)	Menerima bit yang terdapat dalam penjadual terkanan Keluarkan 3 bit pariti
Penimbal 2	Datain (6:0) Dataout (1:0)	Simpan 7 bit input masukan Keluaran sesiri daripada 7 bit input
Litar Pengiraan Pariti	Datain (6:0) Dataout (1:0)	Terima input secara sesiri Keluarkan 7 bit paten ralat

		secara sesiri
Logik XOR	Datain (1:0)	XORkan input masukan dengan bit paten ralat secara sesiri
	Dataout(6:0)	Nilai output sebenar

RUJUKAN

Rujukan : internet

KOD HAMMING

- <http://cegt201.bradley.edu/projects/proj2002/hcedd/seniorprojectpaper.html>
- <http://www.xup.msu.edu/labs/lab1/lab1.htm>
- <http://www.xup.msu.edu/labs/lab2/lab2.htm>
- <http://www.xup.msu.edu/labs/lab3/lab3.htm>
- <http://www.xilinx.com/apps/hdl.htm>
- <http://wwbchat.com/webx?23@198.H3oCacocwLo^0@14%40>
- <http://assembly.nerdworld.com/directory/computers.html>

KOD PEMERIKSAAN LEWAHAN BERKITAR

- <http://www-s.ti.com/sc/psheets/spra530/spra530.pdf>
- http://www.cms.dmu.ac.uk/~se00jf/nets_notes/chunk_html/x521.html
- <Http://www.ece.cmu.edu/~ece548/handouts/20depend.pdf>
- <http://www-s.ti.com/sc/psheets/spra530/spra530.pdf>
- <http://www.santafe.edu/~dsmith/publications.html>
- http://www.oxley.co.uk/pdf/etag/etagapp40604_1.pdf
- http://nicewww.cern.ch/~lebwshop/LEB99_Book/Posters/meggyesi.pdf
- <http://vada.skku.ac.kr/Research/published/vlsi-design2000.pdf>
- http://courses.ece.cornell.edu/ece561/coding_applications.pdf

VHDL

- <http://www.computer.org/cspress/catalog/bp07716/chapt.htm>
- http://www.ee.pdx.edu/~mperkows/CLASS_VHDL/=chapter02.html#T3
- http://www.vhdl-online.de/tutorial/englisch/t_12.htm#pgfId-1003195
- <http://www.ee.princeton.edu/~zhenluo/Pam/synop.html>
- <http://www.connectronics.com/stratum/fec.htm>
- <http://www.eecg.toronto.edu/~de/Cn-edc.pdf>
- <http://www.csc.gatech.edu/~copeland/6092/pdf/slides-08.pdf>
- http://www.wikipedia.com/wiki/Linear_feedback_shift_register

Rujukan : Buku

- Mano M.M and Kime C.R (1997), Logic Computer and Design Fundamentals , pp. 220-231, Prentice Hall, Inc, New Jersey.
- Shu Lin and Daniel J. Costello, Jr (1983), Error Control Coding: Fundamentals and Applications, pp.51-84, 287-313, Prentice Hall, Inc, New Jersey
- Ahmad Ismail (1988), Asas Rekabentuk Logik Komputer, Percetakan Dewan Bahasa dan Pustaka, Selangor darul Ehsan.
- Perry D.G (1991), VHDL, McGraw- Hill, Inc, United State of America.
- S. Yalamanchili (2002), Introductory VHDL: From Simulation To Synthesis, pp. 1- 10, Prentice Hall, New Jersey.
- Bashker J. A VHDL Primer Revised Edition (1994), Prentice Hall PTR, New Jersey.

- Mohd Hisyam b. Abd fatah (2000). *Error correcting & detection code*. Bachelor Thesis. Universiti Malaya.
- Ng Choon Boon (1996/97). *Error correction coding for digital communications*.session on 1996/97. Bachelor Thesis. Universiti Malaya.
- Nurul Amin Badrul (2000). *Error correcting & detection code–A look at BCH codes*. Bachelor Thesis.Universiti Malaya.